IPv6 Rollout To TeliaSonera's Finnish IP-Network

Tero Maaniemi

Thesis
December 2010

Master's Degree Programme in Information Technology
Information Technology

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖN
KUVAILULEHTI

Author(s) MAANIEMI, Tero	Type of publication Master's Thesis	Date 1.12.2010
	Pages 73	Language English
	Confidential ( ) Until	Permission for web publication (X)
Title IPV6 ROLLOUT TO TELIASONERA'S FINNISH IP-NETWORK		
Degree Programme Master's Degree Programme in Information Technology		
Tutor(s) SILTANEN, Jarmo		
Assigned by SOINI, Jyrki (TeliaSonera)		

Abstract

The need for IPv6 addresses is growing rapidly with the increased demand on public IP addresses. The amount of IPv4 addresses is not sufficient to meet that demand. Therefore something has to be done. To get the user to use IPv6 addresses, the services have to be offered with IPv6. For this, the service providers have started to demand IPv6 internet connectivity. TeliaSonera decided to meet this demand by offering IPv6 connectivity to corporate and community customers. This connectivity is provided by 6PE technology.

This thesis focuses mainly on the IPv6 technology, the pilot project and the rollout of IPv6 with 6PE capable edges and routereflectors. Also the outside connectivity with external operators was established.

With 6PE the operator is able to start a smooth transition toward IPv6. 6PE can be enabled over an existing MPLS-infrastructure with minimum investments. This way the operator will be able to provide the needed IPv6 connectivity needed for the proliferation of IPv6 for the service provider.

The implementation itself consisted of network planning, lab testing, a pilot network and the roll-out. 6VPE was not implemented during the writing of this thesis, but it will be studied further within TeliaSonera soon.

Keywords
IPv6, 6PE, 6VPE, MPLS, IP, BGP, TeliaSonera

Miscellaneous

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖN
KUVAILULEHTI

Tekijä(t) MAANIEMI, Tero	Julkaisun laji Opinnäytetyö	Päivämäärä 1.12.2010
	Sivumäärä 73	Julkaisun kieli Englanti
	Luottamuksellisuus ( ) saakka	Verkkojulkaisulupa myönnetty (X)
Työn nimi IPV6 ROLLOUT TO TELIASONERA'S FINNISH IP-NETWORK		
Koulutusohjelma Master's Degree Programme in Information Technology		
Työn ohjaaja(t) SILTANEN, Jarmo		
Toimeksiantaja(t) SOINI, Jyrki (TeliaSonera)		

Tiivistelmä

Julkisten IP-osoitteiden kysyntä on lisännyt nopeasti IPv6-osoitteiden tarvetta. IPv4-osoitteiden määrä ei riitä vastaamaan tähän kysyntään, joten jotain on siis tehtävä. Ennen kuin IPv6-osoitteita voidaan tarjota, on palveluiden oltava saatavilla IPv6:lla. Tätä varten palveluntarjoajat ovat ryhtyneet vaatimaan IPv6-yhteyksiä. TeliaSonera on päättänyt vastata tähän kysyntään tarjoamalla yrityksille ja kunnille Ipv6-yhteyksiä, jotka rakennetaan 6PE-teknologialla.

Opinnäytetyössä keskityttiin lähinnä IPv6-teknologian pilottihankkeeseen ja IPv6:den käyttöönottoon 6PE-kykyisillä reunoilla ja reittiheijastimilla. Hankkeessa toteutettiin myös operaattoreiden väliset Ipv6 reittienvaihdot.

Itse toteutus koostui verkon suunnittelusta, testauksesta, pilottiverkosta ja tuotantoon viennistä.

6PE:n avulla operaattorit voivat aloittaa pehmeän siirtymisen kohti IPv6:ta. 6PE voidaan toteuttaa olemassa olevan MPLS-infrastruktuurin päälle hyvin pienin investoinnein. Tällä tavoin operaattorit voivat tarjota kaivattuja IPv6 yhteyksiä palveluntarjoajille jotka mahdollistavat IPv6:den leviämisen.

6VPE:tä ei toteutettu tämän opinnäytetyön aikana, mutta aiheen tutkimista jatketaan.

Avainsanat (asiasanat)
IPv6, 6PE, 6VPE, MPLS, IP, BGP, TeliaSonera

Muut tiedot

CONTENTS

ABBREVIATIONS

6PE	IPv6 over MPLS
6VPE	IPv6 in MPLS VPN networks
ARP	Address Resolution Protocol
AS	Autonomous System
ASBR	Autonomous System Boundary Router
BGP	Border Gateway Protocol
CIDR	Classless Inter-Domain Routing
CPE	Customer Premises Equipment
GRE	General Routing Encapsulation
IANA	Internet Assigned Numbers Authority
iBGP	interior Border Gateway Protocol
ICMP	Internet Control Message Protocol
IGP	Interior Gateway Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISIS or IS-IS	Intermediate System – Intermediate System
L2VPN	Layer 2 Virtual Private Network
LAN	Local Area Network
LDP	Label Distribution Protocol
LSA	Link State Address
MAC	Media Access Control
MP-iBGP	Multiprotocol iBGP
MPLS	Multiprotocol Label Switching
MTU	Maximum Transfer Unit
NAT	Network Address Translation
NIDS	Network Intrusion Detection System
nPVR	Network PVR
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSS	Operations Support System
PAT	Port Address Translation
PE	Provider Edge
PMTUD	Path Maximum Transfer Unit Discovery
POP	Point of Presence
PVR	Personal Video Recorder
RFC	Request for comments
RIR	Regional Internet Registry
SSM	Source Specific Multicast
TCP	Transmission Control Protocol
TSIC	TeliaSonera International Carrier
VLL	Virtual Leased Line
VoD	Video on Demand
VPN	Virtual Private Network

1 INTRODUCTION

The Internet has grown to be a significant part of everyday life everywhere in the world. Most people have more than one device that uses IP-addresses. In the IPv4 address-space that is now almost completely used, there are 4,294,967,296 possible addresses, from which private addresses account for 17,891,328 and the rest are public addresses. Therefore it seems that IPv4 will not be able to satisfy the need for addresses. The situation now (18.10.2010) with unallocated IPv4 addresses according to IANA is that there is only 5% (that is 214,748,364 addresses) of the total address space available. Sounds like a large number, but with the current rate of usage there will be only 1% of unused addresses left in 2011. The last 5%, from 10% to 5% was used in 9 months.

A great deal has been done to avoid the overspending of the public addresses, the most popular method being NAT (Network Address Translation), where several devices can traverse using only one public address. These setups are becoming more and more limiting when new technologies demand internet access with a device specific public identification. It is said that there are more IPv6 addresses for each square millimeter on earth than there are IPv4 addresses altogether. Another comparison could be that the regular IPv6 block that is assigned to a typical home user is a /64 prefix, which means 18,446,744,073,709,551,616 unique addresses, that is 4,294,967,296 times the total amount of IPv4 addresses.

The use of IPv4 as a protocol is also becoming more limiting when considering mobility, security and multicast. These aspects of IP-traffic have been taken in consideration when designing the IPv6.

This thesis briefly discusses the most significant factors of IPv6 theory. The transition from IPv4 to IPv6 and the techniques behind it are also studied.

The main topic of the thesis is the implementation of IPv6-capabilities in TeliaSonera's core-network, starting with the enabling of 6PE. For this, the topology and architecture has to be studied.

After setting up the topology, this thesis will take an in-depth look at IPv6 and MPLS. Both 6PE and 6VPE were studied. Additionally the use of L2VPN is worth looking into.

Finally, this thesis focuses on the pilot cases and the implementation of the network and topology itself. Processes and OSS-systems are also discussed further in the thesis.

In Order for TeliaSonera to start offering the necessary IPv6 connectivity for its customers, something needs to be done. The first logical step is to offer it to the service providers. This thesis concentrates on different possibilities for achieving this goal.

2 INTERNET PROTOCOL

Internet Protocol or IP is used for transmitting traffic over packet switched network. IP is an intricate part of the total protocol suite which goes by the name of TCP/IP. IP is an Internet Layer protocol in the OSI-model and it is designed to enable packet transfer between a source and a destination based only on the addresses of those two hosts.

The first and still by far the most popular version of this protocol is IPv4. IPv4 addressing consists of four octets and is represented usually in a decimal form xxx.xxx.xxx.xxx where xxx is from 0 to 255. IPv4 addresses were divided into different classes (A through E). These classes became obsolete when classless routing came to be with the introduction of CIDR (Classless Inter-Domain Routing). IPv4 address space consists of both private and public addresses. Private addresses are not routed across the internet whereas public ones are. Today the use of private addresses has increased due to the shortage of public addresses. Unlike public addresses, private addresses are allocated locally and each organization can decide on how to allocate and use them, with some exceptions from the operator side. The downside with private addresses is that, in order to have internet connectivity the addresses have to be translated into public addresses. Usually this means that several private addresses have to be translated into one available public address, which is done by using a mechanism called PAT (Port Address Translation).

On a local or a layer2 network IPv4 uses ARP (address resolution protocol) to bind together the physical address or MAC-address of the networking interface of the destination host to enable layer 2 traffic. ARP is based on broadcast messages. This mechanism is replaced by neighbor discovery in IPv6. Neighbor discovery and its relationship with ARP will examined in more detail in the chapter 3.1.2 Neighbor Discovery and ICMPv6. Also the IPv6 part of the Internet Protocol is studied in more detail later in the thesis.

3 IPV6 THEORY

IPv6 has been in the making for a while now. The main reason for starting the IPv6 development was the realization of the depletion of IPv4 addresses. This time the new address space will be sufficient to serve all imaginable services in the world of data communications. IPv6 differs in many ways from its older sibling IPv4. IPv4 has ARP and ICMP, in IPv6 they have been replaced by neighbor discovery and ICMPv6. In IPv4, multicast was a late developed add-on, and in IPv6 it is one of the features of the protocol architecture.

3.1 General IPv6 functionality

3.1.1 Packet Format and Addressing

IPv6 addresses are 128-bit addresses (whereas IPv4 was 32-bit). The usual representation of an IPv6 address (RFC5952) is a set of eight hexadecimal digits separated with colons ":" (for example: 2001:0db8:600a:0401:0000:8a2e:0370:7304 which then can be shortened by removing the non marking zeros -> 2001:0db8:600a:401::8a2e:370:7304). Figure 1 shows how one address can be represented in 8 different ways (RFC5952, 3).

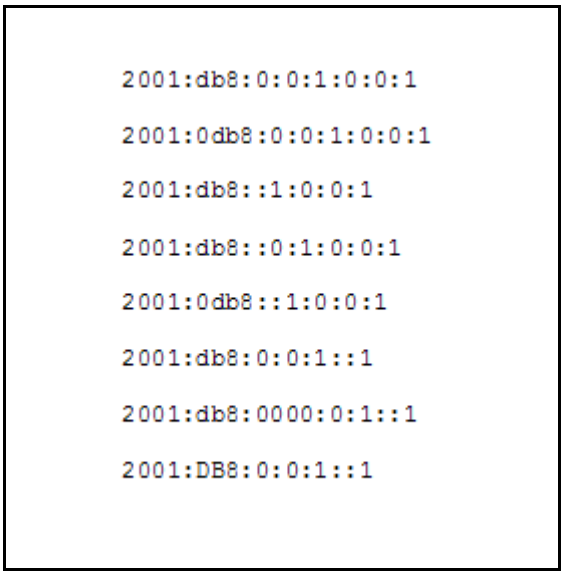


FIGURE 1 RFC5952

The address consists of a 64-bit subnet part and a 64-bit host part, which is usually formed from the MAC-address of the device (48-bits) by stretching the MAC address from EUI-48 (48bit) standard to EUI-64 (64bit) standard. This procedure is described in depth in RFC3513 (RFC3513, 21).

Network masks are not used anymore instead there is prefix length, for example the address block 2001:2000::/20 has been allocated for TeliaSonera and the /20 is the prefix length.

Addresses in IPv6 can be divided into three major types: unicast, anycast and multicast.

“Unicast is a one-to one connection between the client and the server. Unicast uses IP delivery methods such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), which are session-based protocols” (Differences Between Multicast and Unicast). Unicast addresses are used in IPv6 as they are in IPv4, per host. The structure of a global unicast address shown in figure 2 (RFC4291 , 9).

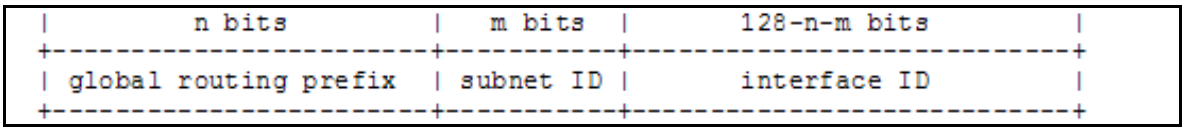


FIGURE 2 RFC4291

Anycast traffic is destined to several interfaces with a common anycast address. This means that the source is not aware that the destination is in fact several destinations. In IPv4 anycast was done by simply assigning the same IP-address for several instances, there were no specific addresses for this purpose. In IPv6 subnet router anycast the only major difference is the interface identifier, which usually is the MAC-address of the interface, will have to be set to zero shown in figure 3 (RFC4291, 13):

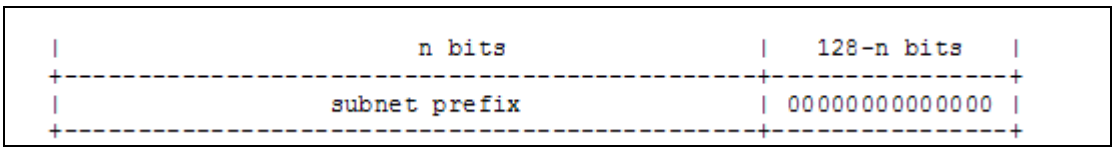


FIGURE 3 RFC4291

One anycast type is anycast embedded-rp, which enables a fast fail-over and shared-tree load balancing with several active RPs in one domain (RFC3446, 2)

There is no need for per host address reservation with IPv6 as there was and still is with IPv4. The IPv6 addresses are allocated so that a /64 is recommended to be allocated for any small subnet (even a p2p-link) A /48 is something that would be allocated for an enterprise, home or a building. ISPs are allocated address-blocks that are /32 or bigger. /128 are used for Loopbacks.

The RFC2460 describes the specifications for IPv6. This RFC concentrates mainly on the packet format of IPv6. As most packets the IPv6 packet consists of a header and the payload. In the case of IPv6 there can also be extension headers which are only read after the packet reaches its destination. These extension headers cannot be found in an IPv4 packet.

Many fields used in an IPv4 header have been left out from an IPv6 basic header. For example internet header length field has been left out and the ToS-field has been

replaced by Traffic Class –field. The differences in these two headers can be seen on In figure 4 (RFC791, 11) and 5 (RFC2460, 4).

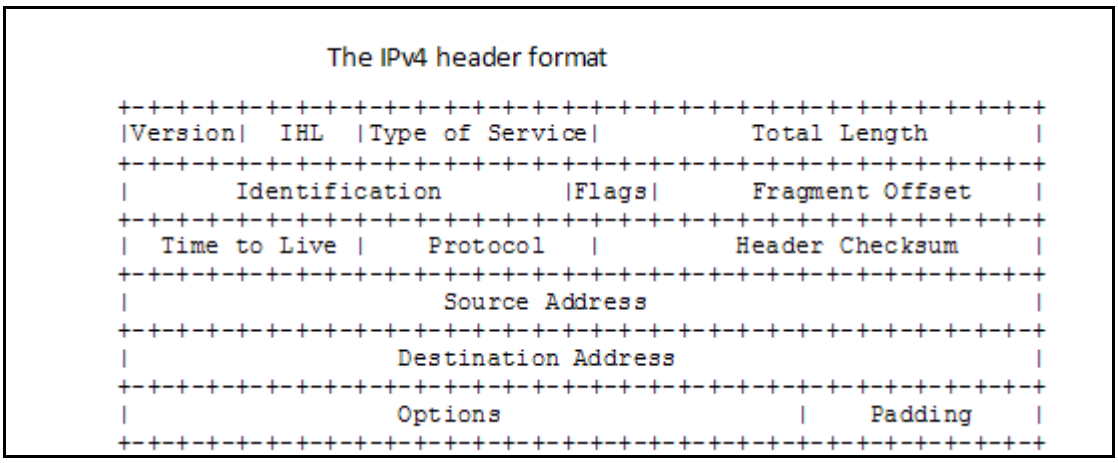


FIGURE 4 RFC791

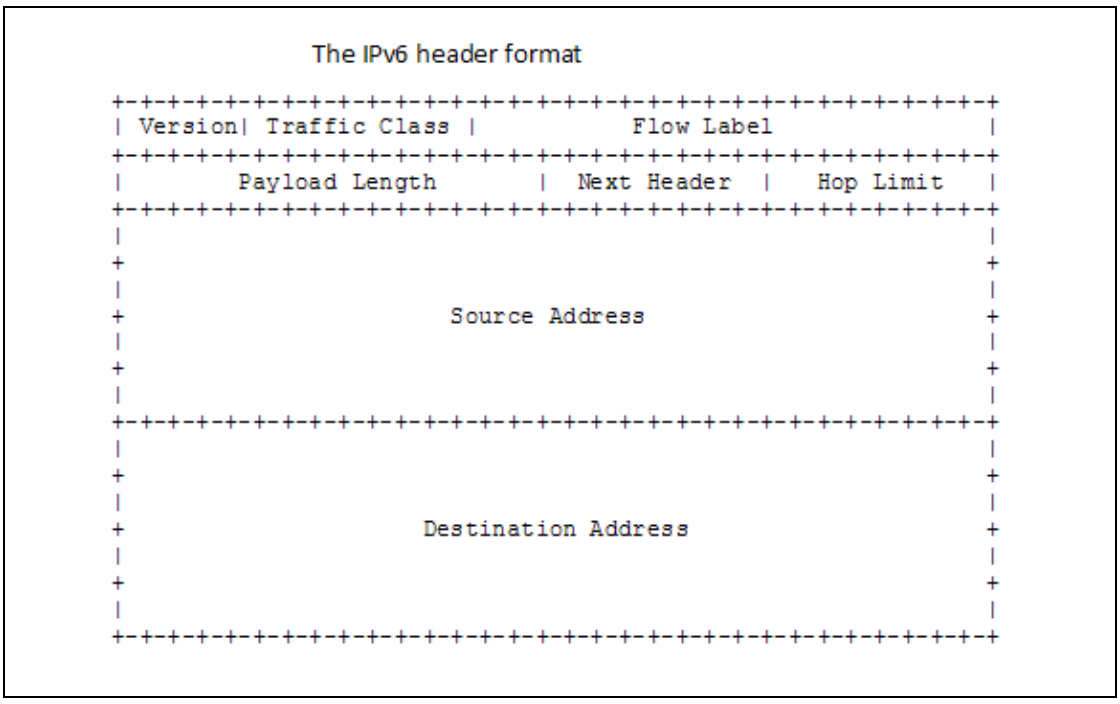


FIGURE 5 RFC2460

The IPv6 header consists of fields mentioned in the figure above. The version field determines if the packet in question is an IPv4 or an IPv6 packet. An IPv6 packet has usually extension headers which determine the options for the datagram. These

headers are: Hop-by-Hop Options, Routing, Fragment, Destination Options, Authentication and Encapsulating Security Payload (RFC2460, 7)

3.1.2 Neighbor Discovery and ICMPv6

All networking devices with Ethernet adapters are equipped with hardware encoded addresses called MAC addresses. These addresses are unique, but cannot be routed over the network. To route the traffic over the network, a translation or a binding between the MAC address and the communication address has to be done.

In IPv4 the MAC address and the IP-address are bound with a protocol called ARP. The ARP uses broadcast to find which MAC-address is connected to which IP-address. For example, when an unknown host is called in the network, the sending device sends an ARP request for all devices connected to the same subnet as an Ethernet broadcast. This request reaches all connected devices and the one that has the correct IP-address answers with an ARP-reply, which then is forwarded to the requester. Then the requester adds this address to its ARP-table and forwards the packet toward the correct MAC/IP combination.

With IPv6 the address resolution is done differently. The concept of broadcast does not exist in IPv6, instead the functionality has been replaced by multicast. The IPv6 neighbor solicitations messages are multicast messages which are forwarded to those hosts that are subscribing to the specific multicast address. The neighbor solicitation starts with a multicast frame sent to a multicast MAC-address which is formed from the IPv6 address in question. With this the host, acquiring an IPv6 address, does not flood the whole network with a broadcast as in IPv4 ARP. Instead the multicast packets are forwarded only to the subscriber of that specific multicast stream.

IPv6 Neighbor Discovery acts differently on different link type, therefore it is wise to go through all link-types and there properties briefly (RFC4861, 8).

A multicast capable interface natively supports the sending of packets to all or a subset of neighbors at the link layer. A point-to-point link two interfaces. A point-to-point interface is assumed to have multicast capabilities, however this is trivial, since there is no need for multicast functionalities on a point-to-point link. Non-broadcast multi-access link is a type of a link where broadcast or multicast capabilities are not natively available. Several interfaces can still be attached to these links on link layer. Examples of such links are Frame-relay, ATM, X25, etc. These links can be considered uninteresting for the scope of this thesis, since these links are used for carrying xDSL traffic over L2VPN-connections in TeliaSonera's network and IPv6 is not used in L2VPN connections. Shared media is a link in which several other nodes are connected to, but are not necessarily able to communicate between each other without traversing through a router. Variable MTU links do not have a well defined MTU (Token ring), compared to Ethernet links which have defined MTU. With Token ring the MTU is defined by maximum token hold time and sending speed. This again is not a problem with TeliaSonera, since Token Ring or other variable MTU link technologies are not offered anymore. The last one is links that have asymmetric reachability. This means that A packet can travel A → B but not B → A or that a packet can travel A → B and B → C but not A → C. Radio links can be considered as such links (RFC2461; RFC4864).

In short the link properties are:

- Multicast capable
- Point-to-point
- Non-broadcast Multi-Access (NBMA)
- Shared media
- Variable MTU
- Asymmetric Reachability

The Neighbor discovery is used for (RFC4861)

- IPv6 address autoconfiguration

- Fetching used prefixes, routes and other configuration information
- Duplicate IP address Detection (DAD)
- Resolving MAC-addresses of hosts on same physical link
- Router discovery
- Neighbor Unreachability Detection

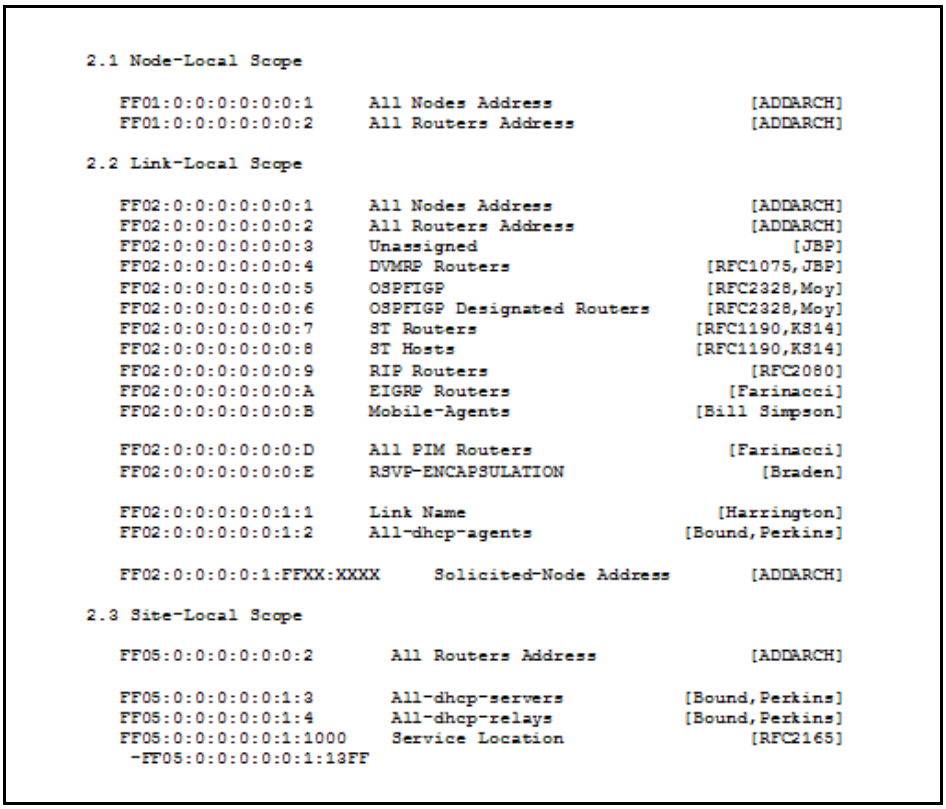
The IPv6 address autoconfiguration can be divided into two parts, stateful and stateless autoconfiguration. Stateful autoconfiguration is done with DHCPv6-servers and does not differ that much from that of IPv4. What is new with IPv6 is the stateless autoconfiguration. With stateless autoconfiguration (RFC2462), the IPv6 address is acquired from the neighboring device by router discovery and router advertisement. Router discovery is either done with router advertisement or router solicitation resulting in the advertisement. Routers send out router advertisements to a multicast address with regular intervals and host connecting to the network listen to that address. Router solicitation (an ICMPv6 message sent to the all routers multicast address FF02::2) is a request sent to routers for them to generate the router advertisement immediately. The router advertisement (ICMPv6 message (RFC4443)) contains options such as MTU, the prefix and the source link-layer address. DAD is used for confirming that all used addresses are unique.

The functionality of ARP requests in IPv4 has been replaced by neighbor discovery messages. A neighbor solicitation message is sent to the neighbor's solicited node multicast address. The neighbor replies to the message with a neighbor advertisement, which in turn carries the MAC-address of the neighbor. This way the messages will only go from host to host unlike IPv4 where all ARP-requests are sent to the gateway or all hosts.

3.1.3 Multicast

Multicast can be simplified as traffic flow which is multiplied on each intersection of the network to all the interfaces which subscribe to the sent out flow. This means that only one flow of information is sent from the source and there is no more than one instance of that flow at any given moment on one link independent of the amount of subscribers. Multicast differs from broadcast in the sense that with broadcast the traffic is sent to all hosts in a given subnet and, in multicast it is sent only to those that have subscribed it.

The concept of broadcast has been put aside in the specifications of IPv6. IPv6 combines the functionalities of multicast and broadcast in its multicast capabilities. The broadcast functionality of sending packets connected to a set of hosts located in the same subnet is implemented in IPv6 by using an address called “All Nodes” in the link-local scope of IPv6 addresses. As with IPv4 the multicast packet is identified by its address (RFC2375). A list of different IPv6 multicast addresses (excluding All Scope Multicast Addresses) is shown in figure 6 (RFC2375, 2).



```
2.1 Node-Local Scope

   FF01:0:0:0:0:0:0:1      All Nodes Address                    [ADDARCH]
   FF01:0:0:0:0:0:0:2      All Routers Address                  [ADDARCH]

2.2 Link-Local Scope

   FF02:0:0:0:0:0:0:1      All Nodes Address                    [ADDARCH]
   FF02:0:0:0:0:0:0:2      All Routers Address                  [ADDARCH]
   FF02:0:0:0:0:0:0:3      Unassigned                              [JBP]
   FF02:0:0:0:0:0:0:4      DVMRP Routers                    [RFC1075,JBP]
   FF02:0:0:0:0:0:0:5      OSPFIGP                          [RFC2328,Moy]
   FF02:0:0:0:0:0:0:6      OSPFIGP Designated Routers       [RFC2328,Moy]
   FF02:0:0:0:0:0:0:7      ST Routers                      [RFC1190,KS14]
   FF02:0:0:0:0:0:0:8      ST Hosts                        [RFC1190,KS14]
   FF02:0:0:0:0:0:0:9      RIP Routers                          [RFC2080]
   FF02:0:0:0:0:0:0:A      EIGRP Routers                     [Farinacci]
   FF02:0:0:0:0:0:0:B      Mobile-Agents                   [Bill Simpson]

   FF02:0:0:0:0:0:0:D      All PIM Routers                   [Farinacci]
   FF02:0:0:0:0:0:0:E      RSVP-ENCAPSULATION                    [Braden]

   FF02:0:0:0:0:0:1:1      Link Name                        [Harrington]
   FF02:0:0:0:0:0:1:2      All-dhcp-agents                [Bound,Perkins]

   FF02:0:0:0:0:1:FFXX:XXXX     Solicited-Node Address          [ADDARCH]

2.3 Site-Local Scope

   FF05:0:0:0:0:0:0:2       All Routers Address                 [ADDARCH]

   FF05:0:0:0:0:0:1:3       All-dhcp-servers              [Bound,Perkins]
   FF05:0:0:0:0:0:1:4       All-dhcp-relays               [Bound,Perkins]
   FF05:0:0:0:0:0:1:1000    Service Location                    [RFC2165]
    -FF05:0:0:0:0:0:1:13FF
```

FIGURE 6 RFC2375

As the Diagram 3.1.3-1 shows the IPv6 multicast address always begins with FF. This translates into binary as 11111111.

The second octet is divided into two 4bit sequences. The first of them is a set of four flags. The first flag is always 0. If the following bit R is set to 1, this indicates that the address is an embedded-RP address (RFC3956). In this case the following bit must also be 1 (R=1 → P=1 → T=1). This will give the address a format of FF7x. The IPv6 multicast address format for embedded-RP is shown in Diagram 3.1.3-2 (RFC3956, 5)

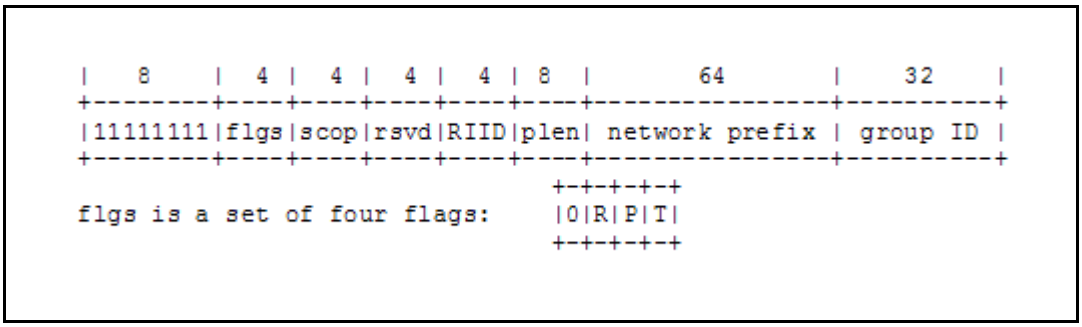


FIGURE 7 RFC3956

If the R is 0 but the P is 1, the address is a multicast address derived from a unicast address (RFC3306). In this case the following T must also be 1 (R=0 → P=1 → T=1). This can be used for example for SSM, but this requires the “plen” (the prefix length) octet and the prefix to be set to 0. This will create an address with the format of FF3x::/32. Address format for a multicast address that is derived from a unicast address is shown in Diagram 3.1.3-3 (RFC3306, 3)

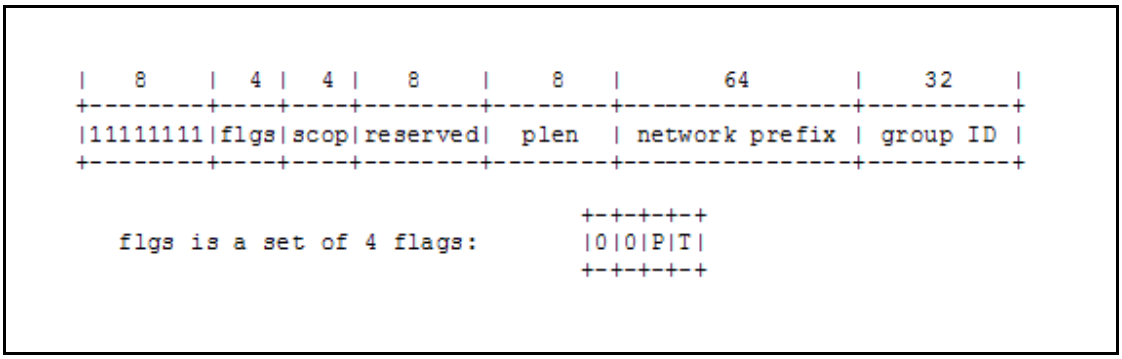


FIGURE 8 RFC3306

Regular multicast addresses are then divided by the last flag into well-known (T=0) and transient (T=1) addresses. The first one is a set of addresses assigned by the global Internet Numbering Authority IANA (Internet Assigned Numbers Authority). The address format for these addresses is as shown in Diagram 3.1.3-4 (RFC2373, 13).

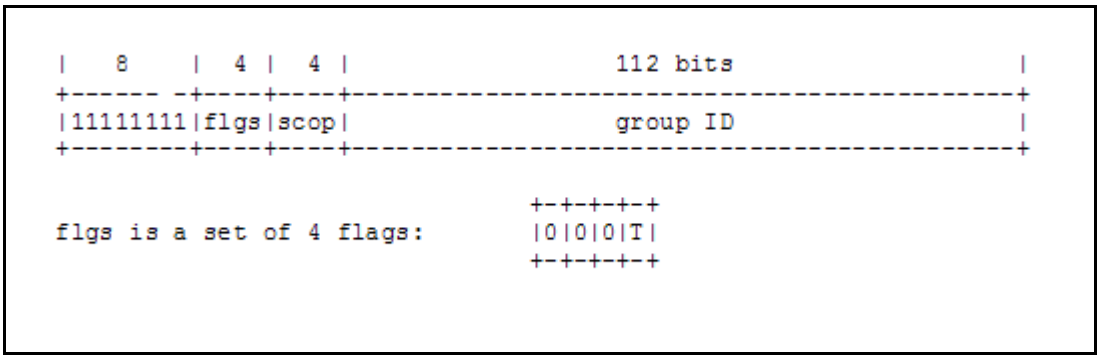
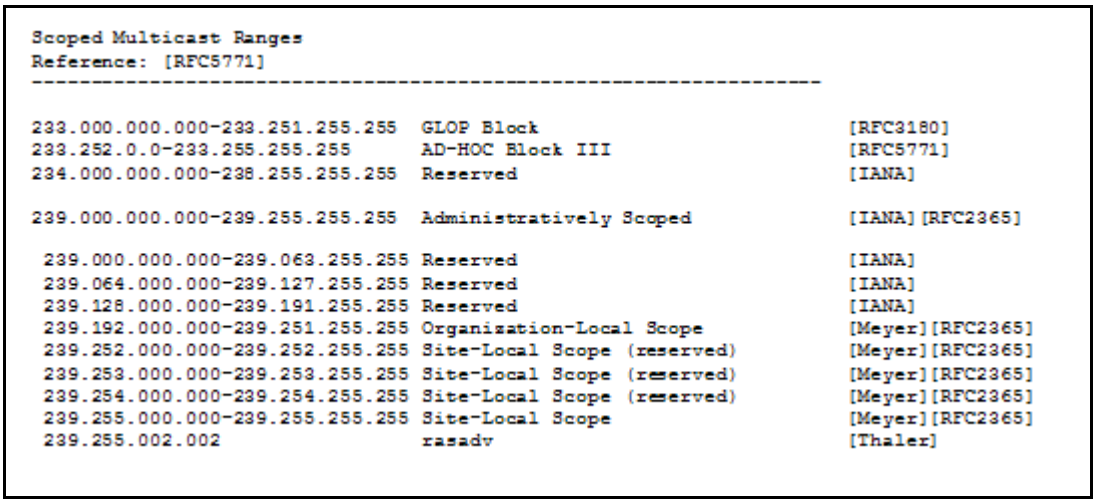


FIGURE 9 RFC2373

With multicast, both in IPv4 and IPv6, there has for sometime been a way of specifying the scope of the multicast "transmission". At first TTL was used for narrowing the scope of spread. With IPv4 the address space is limited and the scopes have been defined as shown in Diagram 3.1.3-5 (IANA, IPv4 Scoped Multicast Ranges)



Scoped Multicast Ranges
Reference: [RFC5771]

233.000.000.000-233.251.255.255	GLOP Block	[RFC3180]
233.252.0.0-233.255.255.255	AD-HOC Block III	[RFC5771]
234.000.000.000-238.255.255.255	Reserved	[IANA]
239.000.000.000-239.255.255.255	Administratively Scoped	[IANA][RFC2365]
239.000.000.000-239.063.255.255	Reserved	[IANA]
239.064.000.000-239.127.255.255	Reserved	[IANA]
239.128.000.000-239.191.255.255	Reserved	[IANA]
239.192.000.000-239.251.255.255	Organization-Local Scope	[Meyer][RFC2365]
239.252.000.000-239.252.255.255	Site-Local Scope (reserved)	[Meyer][RFC2365]
239.253.000.000-239.253.255.255	Site-Local Scope (reserved)	[Meyer][RFC2365]
239.254.000.000-239.254.255.255	Site-Local Scope (reserved)	[Meyer][RFC2365]
239.255.000.000-239.255.255.255	Site-Local Scope	[Meyer][RFC2365]
239.255.002.002	rasadv	[Thaler]

FIGURE 10 IPv4 Scoped Multicast Ranges

In IPv6 the multicast scope is defined also by the address, but with a 4-bit slot reserved for the case. Diagram 3.1.3-6 describes how the scope is defined (RFC2373, 14)

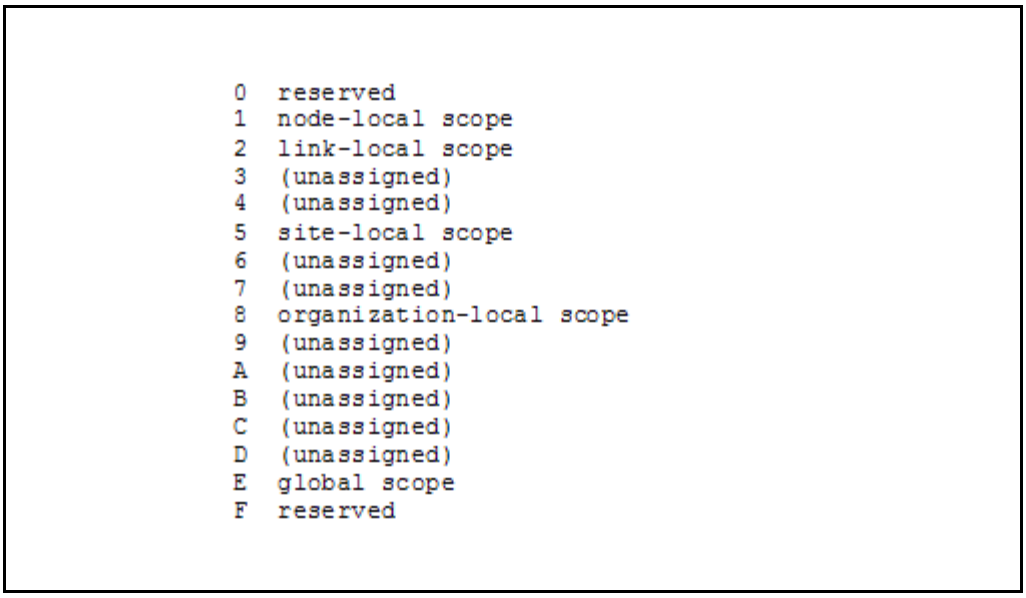


FIGURE 11 RFC2373

With IPv6, multicast can be implemented more easily and effortlessly, and this can be seen as giving multicast a proper chance to spread.

3.1.4 Security

The security issues with IPv4 are mostly relevant in IPv6; the biggest difference again is the amount of usable IP-addresses, which demands a different point of view to the familiar security challenges.

Security threats can be divided into those that have significantly changed with IPv6 and those that have not (2004, S. Convery, D. Miller).

As mentioned, the amount of addresses changes the viewpoint. The most common, probably the most used and generally the first attack is a reconnaissance attack, where the attacker uses such tools as ping sweeps and port scans to find openings in firewalls and active hosts. This is fairly simple in IPv4, but IPv6 it is considerably harder. The sheer size of each subnet makes it impossible to scan. In comparison the

commonly used subnet size in IPv4, one 8-bit c-class (/24) consists $2^8 =$ 256 scannable addresses, while in IPv6 the default subnet is 64-bits (/64) includes 2^{64} = 18446744073709551616 scannable addresses. What took seconds to scan in IPv4 will take years in IPv6. This definitely adds to security. The size itself does not protect known services which have a DNS-name. With IPv6 the DNS will play a significant role, since every service in the network will need a domain name in order for it to be accessible, the addresses themselves are mostly unusable without the names.

As with the reconnaissance, the unauthorized access attacks can also be dealt with differently in IPv6 than in IPv4. With IPv4 the protocol itself does not include and controls to limit access and therefore it has to be limited with access-lists and firewalls. In IPv6 most of the access control will be done with the use of IPsec, which is mandatory in IPv6 (RFC4301).

Header manipulations and fragmentation is used to bypass access-control devices such as firewalls and NIDS. With IPv4 the fragmentations of the packet can be done freely on each networking device. According to RFC2460 “IPv6 requires that every link in the internet have an MTU of 1280 octets or greater” (RFC2460, 24). Additionally, IPv6 uses PMTUD to establish a know MTU for the whole data path (RFC1981). This way none of the devices on the path have to do fragmentation. The fixed MTU set by the sending host eliminates the possibility of using different sized packets to be used in attacks.

Spoofing (both address and port) is commonly used in IPv4 to let the receiving end believe that the traffic in question is coming from another source or another application than it truly is. With IPv4 the address aggregation has been a great deal looser than that of IPv6. With a loose policy with address allocation, it is much harder to prevent spoofing. In IPv6, the addresses are aggregated by default and therefore it will be a lot easier to establish spoofing prevention measures. This of course is only relevant with address spoofing. With port spoofing, IPv6 does not add anymore security and everything that was and is used in IPv4 is still valid with IPv6. (RFC1948)

ARP and DHCP attacks are used to give a client false information of the network. With IPv4 the DHCP-servers are quite commonly used, and all of them are subject to

attacks where a rogue server put into the network can hijack the DHCP-request coming from the clients and then feed them misinformation. The ARP information can be manipulated so that the IP-MAC binding in the MAC-table is inconsistent with the true bindings. This will fool other devices within the L2 area to send information to a correct IP, but the incorrect MAC. In IPv6 the DHCP-service is done mostly with ICMPv6 and neighbor discovery, this includes also the exchange of MAC-addresses which are a part of the IPv6 address. So in most cases IPv6 is safe from these sorts of attacks. The servers themselves still remain vulnerable in IPv6. Since in most cases the router acts as a part of the neighbor discovery, a rogue router can act as one too. This is a vulnerability that concerns mostly IPv6. A rogue router can deliver false network information to a connected host about the underlying network, and this way redirect the traffic falsely.

Broadcast- or “Smurf”-attacks are where the attacker uses a ping send to a broadcast address of the victim network using the victim hosts IP address as spoofed source address. When all the hosts on the network eventually receive the broadcast echo request, they will send the echo-reply to the spoofed address causing it to receive a lot of traffic and in the worst case its service might be denied. This threat has been handled in IPv4 for example with denying directed broadcast. With IPv6 the same problem does not exist, since broadcast is an unknown concept. IPv6 has all-hosts multicast address on a link-local scope, and with use of that a broadcast type attack can be done on a link-local scope with any IPv6 packet for example ICMPv6. It is quite easy to defend against this by implementing a limit for ICMPv6. With other IPv6 packets, this is a bit harder.

Other attacks, such as routing attacks, worms and viruses are handled in the same way in both IPv4 and IPv6.

Some security-problems can be seen in the transition from IPv4 to IPv6 (mechanisms will be gone through later). These concern for example the IPv6 on IPv4 tunneling.

3.1.5 Mobility

With TCP-based applications the connectivity is always based on IP-addresses and ports. If any of these factors change, the connection has to be established again. This is a problem with mobility. When a device is moving from a home WLAN to a cellular network, the infrastructure behind the connections changes and this usually means that also the connecting address changes and therefore all connections have to be re-established. With IPv6 it is possible to move from one network infrastructure to another while holding on to the given IPv6 host identity. This is possible because the IPv6 packet contains a Mobile Header (MH) shown in Diagram 3.1.5-1 (RFC3775, 32).

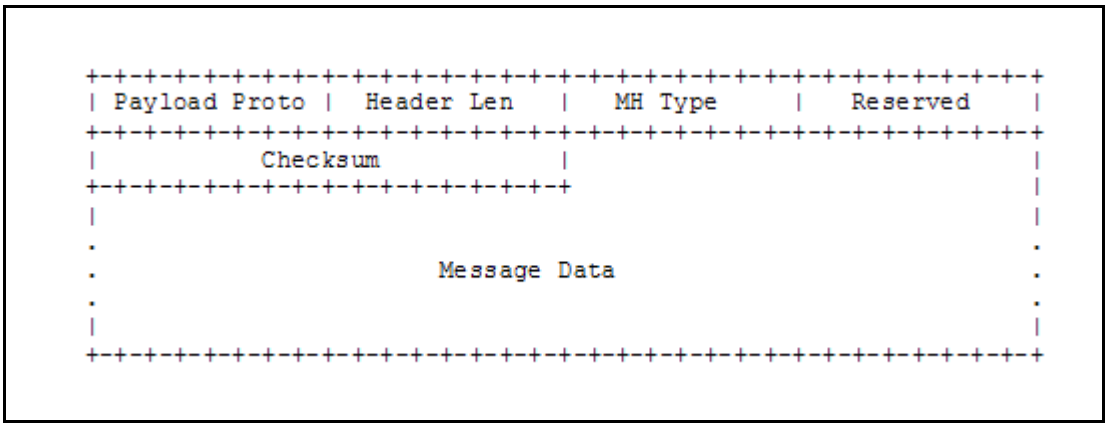


FIGURE 12

The Mobile Header is an extension header for the IPv6 packet. Payload Proto defines the type of the header (in this case Mobility Header) and the Header Len sets the length of the header. MH type describes the type of mobility header in question. With this information the host, node and the network are able to have a continued connection regardless of the infrastructure.

The way that mobility in IPv6 works is that while in the home network the device has a “home address” which is a normally routed address, but when leaving the home network the host, which is also called “mobile node”, receives an IPv6 address by regular autoconfiguration. This address is called a care-of address. With this address

the mobile node then forms a binding with the home router (or home agent). With the binding, the mobile node can either use a tunnel to the home agent or communicate to the other party (correspondent node), or it can also register its care-of address directly with the correspondent node. The first option is called “bidirectional tunneling” and the latter is called “route optimization”.

The bidirectional tunneling can be considered as a straight forward way of connecting, since it does not require any features from the correspondent node. The route optimization will give the dataflow a more optimized route (hence the name). This in turn requires the correspondent node to be capable of handling the IPv6 messages containing the mobility header.

3.2 Routing

Routing in general remains mostly the same between IPv4 and IPv6. This is true as far as the routing protocols in use. This chapter will focus on the differences of IPv4 and IPv6 capable routing protocols (in this case only OSPF and BGP are relevant, ISIS supports IPv6 as is and no differences can be found).

With OSPFv3 the major differences are (RFC2740, 4):

- Protocol processing per-link, not per-subnet (meaning that the communication over an IP-link without a common subnet is possible and therefore the routing protocol should be set per link instead of per address).
- Removal of addressing semantics (No IPv6-addresses are carried in the OSPF packet header, so therefore all the information (For example Designated Routers DRs and Backup Designated Routers BDRs) that was previously associated the IPv4 addresses will now be associated with Router ID)

- Addition of Flooding scope (this addition describes what the scope is on which the LSAs will flooded. These scopes are Link-Local, Area and AS Scope)
- Explicit support for multiple instances per link (multiple OSPF-instances on one link)
- Use of link-local addresses (Link Local Addresses are used as a source address. On virtual links the global scope or site-local addresses have to be used)
- Authentication changes (OSPFv3 relies on the authentication capabilities built in to the IPv6 packet).
- Packet format changes (The OSPFv3 packet is network-protocol independent, lacks authentication and an added instance ID which allows the use of previously mentioned multi-instance links.
- LSA format changes (LSA types renamed and the address information is carried in the payload)
- Handling unknown LSA types (instead of discarding unknown types, as in OSPF for IPv4, they are flooded to link-local scope or stored and then flooded as understood.)
- Stub area support (Mostly retained the same, but the already mentioned handling of unknown LSA-types is added)
- Identifying neighbors by Router ID (no IPv6 information is carried in the header)

There is no BGP specified only for IPv6. It is the same old BGP but with a few additions, for example, the multiprotocol extension that is described in the RFC2858. The multiprotocol extension is implemented in BGP-4 by adding a feature called address-family and subsequent address-family. Address-families specify the used protocols (IPv4, IPv6, IPX etc.) and features (multicast, unicast). The configuration itself can be seen in appendix 2.

4 FROM IPV4 TO IPV6

Since the IPv4 addresses have been proliferated extremely widely across the internet, the transition from IPv4 to IPv6 will be slow. Quite possibly a large number of IPv4 addresses will remain in use for years to come. This is why certain measures have to be taken in order to allow for the use of both address spaces simultaneously. The measures that this thesis will go through are Dual-Stacking, Tunneling, Proxying and Protocol Translation.

4.1 Dual-stack

The idea in dual-stack is that a node supports the features for both IPv4 and IPv6. This sort of node can be used on the border of IPv4 and IPv6 networks. This node can be either a router or a host. A host running dual stack requires an IPv4/IPv6 capable DNS resolver. (Hagen 2006, 255)

In some cases the operator needs to have a dual-stack network, or at least a partly dual-stack network. This means that all routing protocols must support both protocols. With OSPF, the network would have to be configured with OSPFv2 and v3, even though the OSPFv3 should support IPv4 according to standards it is not supported by major vendors yet. This is extremely hard to handle properly and therefore ISIS is more suitable for dual-stack networks, since it supports both.

4.2 Tunneling

The IPv6 traffic can be forwarded through a native IPv4 network by tunneling or encapsulating the IPv6 traffic in the header of the IPv4. This way the IPv6 packet flows through the network inside an IPv4 packet.

There are several ways of encapsulating/tunneling the IPv6 traffic: Manual IPv6 tunnels, IPv4-compatible tunnels, GRE, Automatic 6to4 tunnels, Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) Tunnels (Hagen 2006, 256) .Worth mentioning are also Teredo tunnels and 6rd or IPv6 Rapid Deployment.

Manual tunneling is what it says it is. It is a tunnel where the end points are configured manually to a router on IPv4 and the tunnel interface on each end has an IPv6 address.

GRE tunneling, described in RFC2784 [12], describes a general tunneling mechanism that was first proposed to be used for transporting “IP over IP for policy purposes” (RFC2784). GRE tunnels are point to point links as the manual tunneling is. GRE is an independent carrier protocol, which can be used for carrying not only IPv6, but also IPv4, CLNS, IS-IS etc, which will in turn be passenger protocols. If the GRE tunnel is an IPv4 GRE tunnel, the end routers must be running Dual-Stack to be able to handle both IPv4 and v6.

6to4, described in RFC3056 [13], is a mechanism where the IPv6 address is formed in the way that it includes a global IPv4 address. The sites where these sorts of addresses are used are called 6to4 sites. The router on the border of the area is a dual-stack router. When two hosts are connected to separate 6to4 sites with IPv4 connectivity in between, the sending host’s destination address contains the IPv4 address of the other areas border router. With this information, the border routers are able to create an automatic tunnel between the routers. In the case of 6to4 to native IPv6, a relay router is used the act as a 6to4 tunnel end-point. The prefix used

for this is 2002::/16, and the IPv4 address will be embedded in the address after the first two octets.

ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) tunnels do basically the same as 6to4, but with ISATAP the use of private IPv4 addresses is possible. As the name suggests, this protocol can only be used within a site and not between sites. ISATAP is described in RFC5214[16] http://www.ietf.org/rfc/rfc5214.txt (which obsoletes the RFC4214.

On top of the router based tunneling mechanisms that were described in this, there is a possibility of using a client/server based solution called Teredo tunneling. Teredo tunneling was established to enable the NAT traversal for IPv6 packets encapsulated in an IPv4 datagram. This helps if the NAT device is not capable of handling 6to4. With Teredo tunneling, the host is connected to an IPv4 network and connects to a Teredo server to establish a tunnel through the IPv4 network. Addresses used in the Teredo are form the prefix 2001:0000::/32. The IPv6 packets are encapsulated in IPv4 UDP packets as payload and therefore the IPv4 network in between the server and client do not have to be aware of the IPv6 content. (RFC4380)

4.3 Proxying

With proxying, the dual-stack feature that was mentioned in the preceding chapters, is done on the application layer by a proxy server, which is needed if the upper level protocol includes the host IP (for example FTP and SIP). This functionality can be implemented in any server, for example a web-proxy, SMTP-server, ftp-server etc. The point is to allow the use of IPv6 and IPv4 on an application level. A general term to describe these servers is the ALG of Application Layer Gateway. (RFC2663)

4.4 Protocol Translation

With IPv4 the use of private addresses demanded an address translation from private to public addresses. The same mechanism can now be used to translate IPv6 addresses to IPv4 addresses. With private IPv4 addresses, several private addresses could be translated into a fewer set of public addresses by using different port numbers. For example if the addresses 10.10.10.1 and 10.10.10.2 were to communicate over the NAT (network address translation) device with the same port number and the translation were to be done to one public address, the communication from the two different hosts would be translated into two different port numbers and one common address.

With IPv6, the address space is significantly larger than that of IPv4, therefore the same mechanism would have to be used. This mechanism is also called NAPT (Network Address and Port Translation).

For use in IPv6 a separate RFC has been written. The NAT-PT or Network Address Translation – Protocol Translation is described in RFC2766. RFC has been obsoleted by RFC4966[18], which basically describes the reasons why this mechanism cannot be thought of as a general purpose transition mechanism.

DNS64 and NAT64 are used for connectivity between IPv6 and IPv4 networks. These two services or servers are used together, but do not share states. The process of handling IPv6 to IPv4 connectivity is as follows:

1 IPv6 host sends a DNS query to the DNS64 server.
2 DNS64 server returns a IPv6 address which is formed from the address of the NAT64 servers Pref64 pool and the end host IPv4 address. For example Pref64 is 2001:db8::/96 and the IPv4 end host address is 10.10.10.10 the returned address is 2001:db8::10.10.10.10. Which is in hexadecimal 2001:db8::aaaa
3 IPv6 send the datagram to the end host 10.10.10.10 with a destination address of 2001:db8::10.10.10.10. Which is in hexadecimal 2001:db8::aaaa

4 NAT64 translates the source host address to IPv4 and strips the IPv6 part from the sent datagrams destination.

The functionality of these services is described in two IETF drafts: Stateful NAT64 and DNS64.

5 NETWORK TOPOLOGY AND ARCHITECTURE

Network topology and architecture can be considered as a framework set to control the network build-out. In this section the models are studied to setup the framework.

5.1 Topology and Architecture theory

5.1.1 Topology Models

The topology should be viewed as a combination of physical and logical topology. The physical topology is set by point-to-point connections between nodes and is most likely a combination of several basic topology models: ring, tree, bus, star and mesh shown in figure 13. (2008, A Guide to Network Topology)

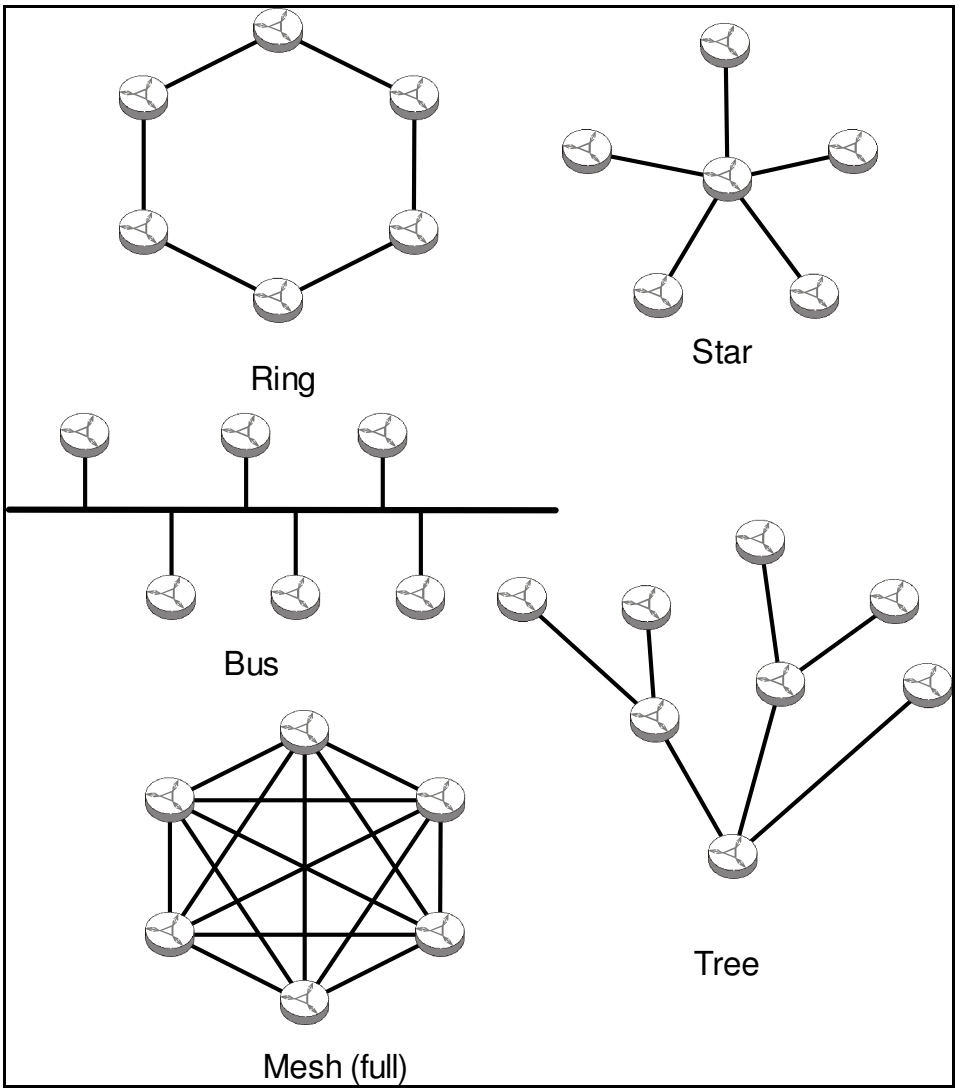


FIGURE 13 Basic Topology Models

In an IP-Core network the bus topology could be considered an unviable choice, however, all other topology models are usually seen in larger networks.

In a ring topology all nodes are connected to two other nodes forming a ring where one link break never causes a total outage. This topology is a redundant topology, but when the traffic flows are not equal on all nodes and, for example one node is providing connectivity to services used by the user behind all other nodes, the traffic will strain the nodes that are between the user and the service.

In a star topology all nodes are connected to one central node. This, compared to the ring topology, is optimal when considering bandwidth load, if all services are also

centered in the central node. Redundancy, on the other hand, is poor to say the least. If the central node fails the whole network fails.

The tree topology is closely related to the star topology; in fact it is an extended star where the failing of the central node causes the branches to lose connectivity with each other, but not the connectivity within each branch. Also a tree topology will ease the construction of a network that is spread over a wide area (such as Finland).

The best topology for redundancy and latency is always a mesh topology, since by definition everything is connected to everything; however, there are a few problems with this sort of topology. Usually it is too expensive and the routing and capacity management is extremely hard to handle (unless the network is built with maximum capacity everywhere and the delays on the links are always the same).

5.1.2 Network Topology in Practice

In this thesis the concept of logical topology is mostly viewed as BGP-topology and route-reflector topology, since the ISIS topology does not play a major role in the roll-out.

The global internet topology consists of autonomous systems which have a peering connection with each other. Larger operators usually have several autonomous systems (later AS) usually due to the need to implement a consistent routing policy and in some case due to the need to differentiate parts of the network. These autonomous systems are usually too large for the joining node to have a full-mesh style of peering. In these situation route-reflectors are used to give the illusion of a full mesh. In very large networks the use of simple route-reflectors or clusters of route-reflectors is not enough. For more complex and larger networks a concept called a confederation is needed. Confederations preceded route-reflector topologies. In a confederation the huge public AS is split into several sub-ASs which

then have a eBGP peering to the public AS, so there are ASs with route-reflectors within an AS. Route-reflectors within an AS can be connected together in the same fashion as the physical topology. All basic topology-models can be used.

5.1.3 Device Roles in Architecture

A modern MPLS-network consists of routers and switches which all have their roles in the network. The core is used for forwarding label switched traffic and nodes, in the role of just forwarding packet are called P- or provider-routers. The role of the P- or provider-router can be divided or to different devices, one for providing bandwidth and one for providing port capacity for edges. These routers should not take part in BGP-routing, meaning that the core network should be BGP-free. The routing decisions should be done in the PE- or provider edge routers. These routers are used for aggregating end-user connections or CPEs (customer premises equipment). Route-reflectors are used for providing routing information for PEs. In a physical topology a route-reflector is located in the topological level with the PEs.

An access-network is usually used for connecting to the PE-routers. These networks are usually L2 networks, which can be anything from large L2VPN-networks to simple switches. In some cases straight connections to PE routers are used, but since a port on a BGP/MPLS capable edge-router costs more than a port on a Metro-Ethernet switch, this is not advised.

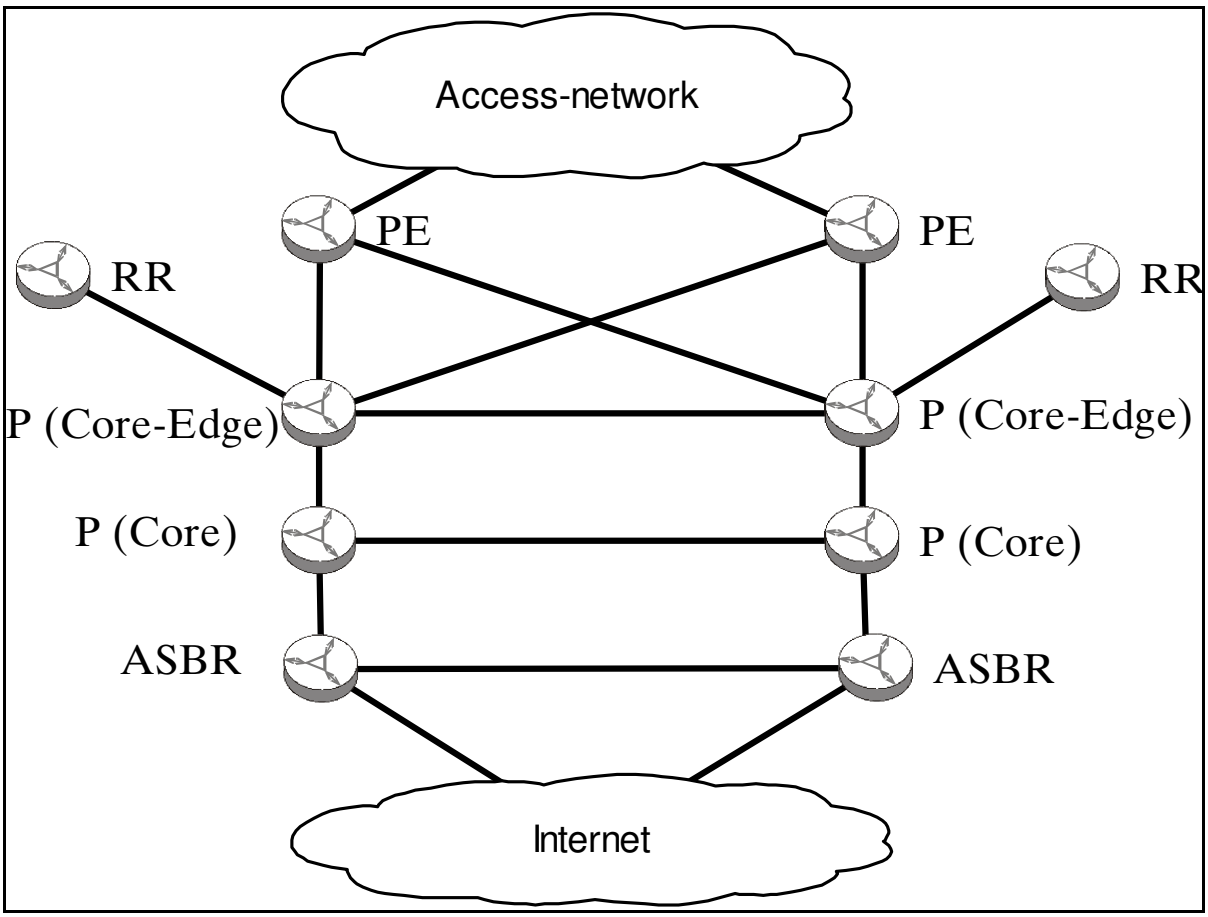


FIGURE 14 Physical Topology Model

A few network specific devices should be mentioned here. One is the ASBR, which is used for separating the core network from the AS to AS connectivity and freeing the core from BGP-routing. This device can give the network that it is serving more security and better flexibility in bandwidth management since inter-AS traffic is separated into a dedicated device. The term ASBR originates from OSPF and is not considered a mandatory device in a model topology. Some other devices that are not considered mandatory, but still are very useful, are for example the traffic analyzers, guards and sinkholes.

5.1.4 Operators Role

Basically the operator's main role is to provide the needed connectivity for its customers according the regulations. To do this an operator needs connectivity to outside operators. This connectivity is provided through both commercial transit operators and non-commercial Internet Exchange points (IX or IXP) (Global Internet Exchange Points). Finland hosts three non-commercial peering points, Ficix1 (Helsinki), Ficix2 (Espoo) and Ficix3 (Oulu) (FICIX). Traffic through these points is governed by public rules, which each operator sets, and these are called a routing policy. These rules can be found by searching the operator's AS-number in the regional internet registry (RIR) database. In Europe this is RIPE NCC (Halabi, S. 2001. 27).

An operator usually has also regulatory requirements from governmental regulators (In Finland this regulator is Ficora (FICORA)). These requirements have to be met in order to maintain the legal status and the permit to operate as a telecom operator. These regulatory requirements usually describe issues like storing and handling identification data or service availability in terms of service hours, geographical reach and bandwidth. With IPv6 there are no specific regulatory demands as of now (12/2010); however, but some demands can be anticipated.

5.2 TeliaSonera Topology and Architecture

As all prominent operators, TeliaSonera's network topology is in a constant state of change. While building the new architecture the old architecture has to be maintained and migrated in to the new architecture. The following is the best estimate where this topology now is and what could be done to improve it.

5.2.1 Present Topology

The basic network topology of TeliaSonera Finland follows mostly the figure 15. The network consists of 5 POPs (Point of Presence) that can be considered the Core-POPs. The Core-POPs are points where the traffic is transmitted through and which are not used only for network aggregation, but also for transit between several POPs. (2008, Maaniemi)

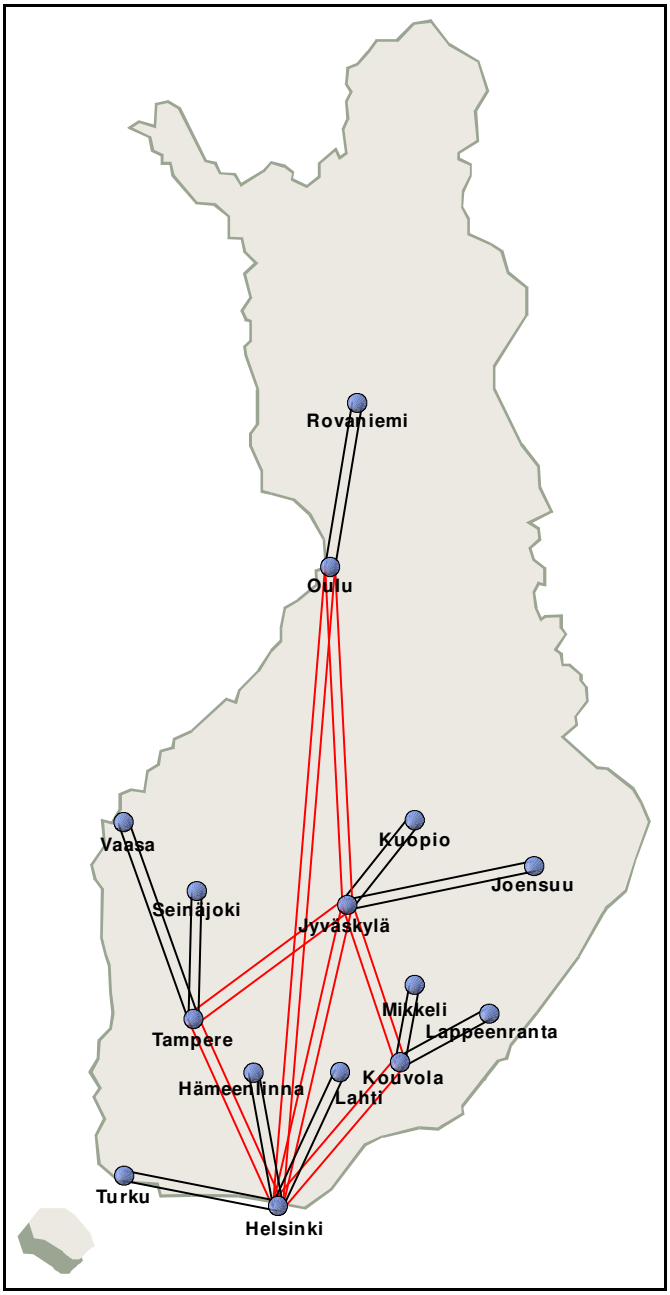


FIGURE 15 (Appendix 5)

5.2.2 Suggestions for Future Topologies

The basic core topology will quite probably follow the present topology for a long time due to geography. The only major change might come from the VoD, nPVR and other bandwidth consuming operator services. These services cannot be handled only in major POPs, and therefore they have to be spread out inside the access-network. This way the stress on the access-network is lowered. For example, if a VoD-server is located in a major POP, all the users within the POP's area will stress the last L2 link within the major POP. This is why the traffic should be stopped by having the first L3 link nearer to the customer.

In logical topology, the more efficient use of several geographically spread AS-border routers might have to be taken in consideration.

Already we can see a change in general topology from a star topology to a partial mesh topology, with the use of east to west cross-links. In the future the capacity of those links will have to be equal to the present main links.

6 IPV6 AND MPLS

As most operator networks, TeliaSonera's network is based on an MPLS Backbone. This simplifies the roll-out of IPv6 in TeliaSonera's network, since the underlying network does not need to be IPv6-aware in order to carry IPv6. With MPLS, the routing (or if it can still be called that) of packets is based on labels. The IGP routing in TeliaSonera's network is done with ISIS, and at this moment does not have to carry IPv6, since all network facing interfaces are still IPv4.

6.1 6PE

6.1.1 What is 6PE

"6PE is the Cisco name for directly carrying IPv6 packets over the MPLS backbone." (2007, De Ghein). 6PE can be considered the first step for a MPLS-capable operator on the road to using IPv6 commercially. 6PE needs IPv6 capable BGP called MP-iBGP to function in a 6PE solution all PE routers are running dual-stack (mentioned earlier), and with 6PE the IPv6 networks are not in an MPLS-VPN and there are on IPv6 VRFs. IPv6 in MPLS-VPN is called 6VPE.

According to RFC4798 6PE can be enabled in an MPLS network that is already able to run BGP/MPLS IP VPN services. From this assumption the 6PE is in a configuration point of view a set of dual-stack PE-routers connect together by BGP over an MPLS-network. The MPLS network does not have to be IPv6 aware in any sense. The BGP peering could be done from PE to PE, however, in operator networks, the use of route-reflectors is recommended. The BGP-peering is done with IPv4 addresses, since they are routed via the IGP.

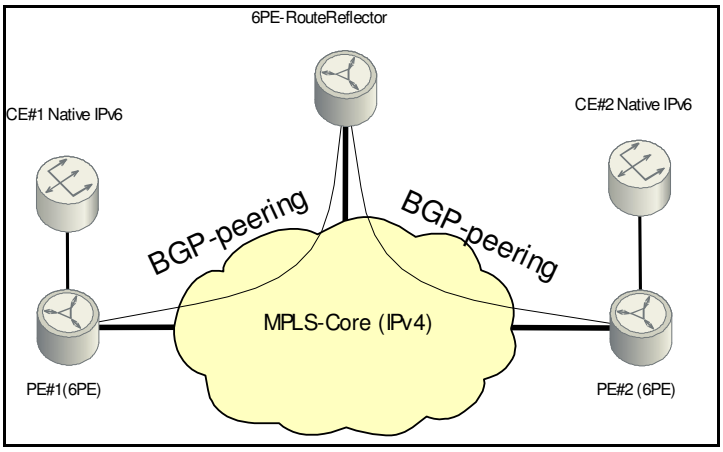


FIGURE 16 MPLS/BGP-topology

6.1.2 Packet Format and Labeling

The major advantage over other tunneling protocols in using 6PE is the labeling. Instead of adding a header to the IPv6 packet by for example encapsulating the packet, in 6PE the IPv6 packet is simply labeled as shown in Figure 17.

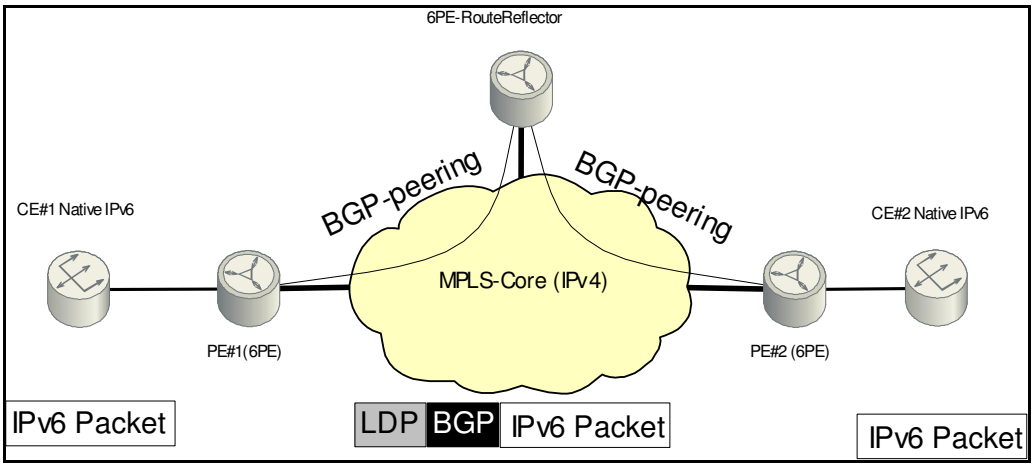


FIGURE 17 Packet format

Only the LDP information is relevant for the underlying MPLS network. Route-information is obtained by BGP. 6PE uses the BGP to carry the extra label for the IPv6 prefixes.

6.1.3 Pros and Cons

The 6PE is easy to adopt in an MPLS-capable environment. As already mentioned the only feature needed is BGP-capable dual-stack PE-routers. With 6PE routing toward the CE can be done with any IPv6 capable routing protocols and with 6PE there is no overhead in the packet. The only drawback, on top of not being able to forward multicast, might be the inability to recognize the IPv6 routes within the MPLS-core in a troubleshooting situation, but this cannot be considered a major drawback, since the labeling can be traced back and therefore the connection troubleshooting can be done. Besides, the MPLS network is not supposed to relay all the routing information, but only that which is needed for label distribution. So the IPv4 addresses, if redistributed in the same fashion would not be traceable either.

6.2 6VPE

6.2.1 What is 6VPE

6VPE is 6PE, but with the ability to carry MPLS-VPN information. The use of MPLS-VPN is mostly the same in both IPv4 and IPv6. The only differences are in the configuration and the VPNv6 and vpnv4 prefix formation. With IPv4 the VPNv4 label is formed with, according to RFC4659 the IPv4 address and the route distinguisher (RD). With IPv6 the VPNv6 prefix is similar, with the IPv6 address and the RD. Since the packets will be routed over the IPv4 MPLS-backbone, the next-hop addresses in vpnv4 on the BGP-table are the loopback addresses of the end hosts which are IPv4 addresses. With vpnv6 the BGP-table naturally consists of IPv6 addresses, therefore

the vpnv6 next-hop is formed by adding “::FFFF:” in front of the IPv4 loopback address.

6.2.2 Packet Format and Labeling

The 6VPE packet format is basically the same as the 6PE packet format, but instead of carrying the IPv6 info, it carries the vpnv6 info

6.2.3 Pros and Cons

The 6VPE is easy to adopt in an MPLS-capable environment. As already mentioned the only feature needed is BGP-capable dual-stack PE-routers, with MPLS VPN -capabilities. With 6VPE routing toward the CE can be done with any VPNv6 capable routing protocols and with 6VPE there is no overhead in the packet. The only drawback, on top of not being able to forward multicast, might be the inability to recognize the VPNv6 routes within the MPLS-core in a troubleshooting situation, but this cannot be considered a major drawback, since the labeling can be traced back and therefore the connection troubleshooting can be done. Besides, the MPLS network is not supposed to relay all the routing information, but only that which is needed for label distribution. So the IPv4 addresses, if redistributed in the same fashion would not be traceable either

6.3 Using L2VPN

6.3.1 What is L2VPN

L2VPN or Layer 2 Virtual Private Network is used for connecting two sites to each other on the OSI layer2 and therefore enabling a common layer3 network between the sites. L2VPN services are either point-to-point, hub-and-spoke or full-mesh. L2VPNs are tunnels that run on top of an MPLS-network. (Hellberg, C., Greene, D. & Boyes, T., 2007, 128)

6.3.2 Packet Format and Labeling

L2VPNs by definition are VPNs running on the OSI's layer2, where the layer3 packets are encapsulated into usually Ethernet frames which are then forwarded over the L2VPN network. This way the L2VPN network is unaware of the upper layer routing and protocols.

6.3.3 Pros and Cons

L2VPN allows a flexible use of any upper layer protocol; however, it is more restricted compared to regular routed network. With an L2VPN network all hosts needing connectivity, must be configured to be connected beforehand. For restricted networks the L2VPN solution is a viable choice.

7 IPV6 PILOT AT TELIASONERA FINLAND

7.1 Initial Network Structure

7.1.1 Route-Reflectors

Route-reflectors are used for route-distribution in medium sized BGP-networks. The route-reflector will act as a hub for BGP-neighbors in the same AS (autonomic system). If route-reflectors were not used, all actors in the same AS would have to be peers with each other. The amount of neighbors will increase by a factor of n*(n-1) compared to the amount of nodes. It was decided to separate the IPv6 functionality from the rest of the IP-core BGP by providing a dedicated route-reflector. This topology provides more stability for the remaining IPv4 network and more flexibility for the IPv6 route-reflector and there software versions.

Two route-reflectors were set up for IPv6 use. At first these route-reflectors were connected to the lab network AS65100 for testing purposes. When the testing was done the peering was switched over to our core AS1759.

The peering toward 6PE routers was done using IPv4 as peering address, since the IPv6 addresses are not yet routed in TeliaSonera’s network. The route reflectors themselves peer toward TeliaSonera’s AS-border where the IPv6 operator peering is done.

7.1.2 PEs

PEs that are used in the pilot are dedicated service routers providing internet connectivity for large corporate and business customer. These PEs will enable the use of public AS-numbers and IP-transit for customers over TeliaSonera's network to its peering partners in both FICIX and TSIC (TeliaSonera International Carrier). MPLS-VPN features are not enabled since they are offered in other service-edges dedicated for that use.

The PE-devices used in the pilot are Cisco 7600 series routers located in Helsinki and Tampere. All policy-maps used for IPv4 (mostly following the Finnish routing-policy) have been reproduced to work with IPv6.

Link-address blocks with IPv6 are pre-assigned and dedicated for each individual PE. This way the link-addresses of each PE can be identified in the routing-tables and fault-management will be easier. Additionally there is no need for separate address-allocation[7]. Link-addresses are allocated according to Metro-Ethernet area or service shown in figure 18.

DNS	2001:2000:6000::/48	Other infrastructure services	2001:2000:6001::/48
Helsinki	2001:2000:600a::/48	Lappeenranta	2001:2000:6014::/48
Loopback	2001:2000:6018::/48	Tampere	2001:2000:601e::/48
Turku	2001:2000:6028::/48	Jyväskylä	2001:2000:6032::/48
Seinäjoki	2001:2000:603c::/48	Vaasa	2001:2000:6046::/48
Oulu	2001:2000:6050::/48	Kouvola	2001:2000:605a::/48
Lahti	2001:2000:6064::/48	Kajaani	2001:2000:606e::/48
Joensuu	2001:2000:6082::/48	Mikkeli	2001:2000:6096::/48
Rovaniemi	2001:2000:60a0::/48	Kuopio	2001:2000:60aa::/48
Pori	2001:2000:60f0::/48	Kotka	2001:2000:6022::/48 (290=0x122)

FIGURE 18

Within these address blocks the link-addresses are then allocated to each router in smaller /56 blocks. From there forward each interface will be given a /64 block for multi-access and /126 for a point-to-point link. This means that each router can have a maximum amount of 255 multi-access interfaces and an unlimited amount of point-to-point interfaces. Loopback addresses are allocated from a separate block

mentioned above. The last 6-octets of the addresses are formed using the BGP router-id making each address a /128 block.

7.2 Blackholing/Sinkholing

Blackholing is a method to drop all incoming traffic towards a target under attack on the network ingress points, thus preventing it from reaching the target and overloading TSF infrastructure elements within its path to the target.

Sinkholing is a method to redirect all incoming traffic for certain target towards a specific Sinkhole router, thus preventing it from reaching the target. It is possible to analyze the traffic on the Sinkhole router itself (e.g., Netflow) or utilize other means (e.g., analyzers, graphers) connected to the Sinkhole router. (2007, Suontausta)

Every router hosting IPv6 will have static route for a chosen blackhole address (2001:2000:6018:ffff:ffff:ffff:ffff:ffff/128) which will point toward Null0. All traffic that is routed toward that address will be discarded by each PE. By default the route-reflector advertises certain routes with a next-hop of the black-hole.

prefix	description	next-hop
::/0	default	2001:2000:6018:ffff:ffff:ffff:ffff:ffff/128 (blackhole)
2001:2060::/32	Customer Network	2001:2000:6018:ffff:ffff:ffff:ffff:ffff/128 (blackhole)
2001:2000:6000::/40	Link Network	2001:2000:6018:ffff:ffff:ffff:ffff:ffff/128 (blackhole)
2001:2001:6000::/40	Test Network	2001:2000:6018:ffff:ffff:ffff:ffff:ffff/128 (blackhole)

FIGURE 19

These addresses are routed to each PE

7.3 Internet Connectivity and DNS

7.3.1 Peering

BGP-peering from the operator's point-of view can be divided into three areas. One is the operator peering done to other internet operators; the other is the one which is done toward TeliaSonera's customers. In all of these cases TeliaSonera already supports IPv6. One is the iBGP peering toward the route reflectors which was already covered in a previous chapter.

The external peering in TeliaSonera's AS1759 is done toward TSIC (AS1299) and Ficix (several AS's). The national transit traffic is dealt with the Ficix peering partners and all other external traffic is routed toward AS1299. This separation is done by local preference in BGP. The peering toward Ficix-partners is carried out with a higher preference than the peering toward AS1299. This way the prefixes, which are announced by customers peering toward both a Ficix partner and AS1299, will be routed toward the Ficix link and will stay in national routing. All transit traffic will in any case go to the AS1299 peer, since none of the Ficix partners allow transit through their network from other Ficix-partners. As shown in Figure 20.

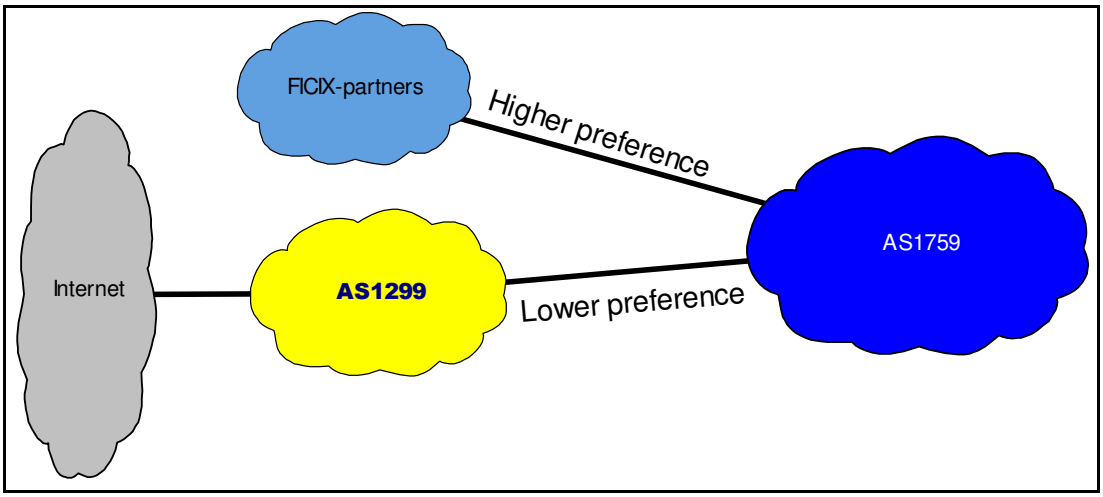


FIGURE 20 Peering preference

The customer peering with IPv6 is limited to 6PE capable routers at first. This is done using the same routing-policy which is in place for IPv4 routing. This routing policy can be found at RIPE Network Coordination Center at www.ripe.net shown in figure 21.

```
remarks:    ============================================================
remarks:    Community support for transit customers
remarks:    ============================================================
remarks:
remarks:    1759:1050   Set local pref 50 (lower than upstream LP)
remarks:    1759:1110  Set local pref 110 (lower than CIX LP)
remarks:    1759:1150  Set local pref 150 (lower than primary customer LP)
remarks:    1759:666  Black hole routing
remarks:
remarks:    Prepend x times (x=1,2,3) or do NOT announce (x=0)
remarks:
remarks:    1759:100X All direct peers
remarks:    1759:200x Upstream Provider
remarks:    1759:300X Cix and private peers
remarks:    1759:400x CIX
remarks:    1759:500x Private peers
remarks:
remarks:    ============================================================
```

FIGURE 21 RIPE AS1759 routing policy

Each customer, with their own CPE, can use all communities mentioned above to change the routing of their prefixes from and to AS1759. (AS1759 Routing Policy)

7.3.2 DNS

DNS servers will work as dual stack servers. Due to the nature of this thesis, the configuration of the DNS servers is not relevant. The servers themselves are crucial for the use of IPv6 and a significant part of the implementation of IPv6 service.

7.4 Pilot Cases

The first pilot cases were mostly large corporations which have their own CPEs. In these cases the configuration in the customers end is naturally the customers' responsibility. There were a few pilot cases where the CPE was in TeliaSonera's area of responsibility.

7.4.1 Configuration

The pilot configurations can be found in the appendixes. The configuration consists of three parts: the PE configuration (Appendix 2), RR configuration (Appendix 1) and the CPE configuration (Appendixes 3 and 4). Configuration is stripped of all unrelated information and all IP-addresses are changed.

The first pilot case was a one CPE internet connection with static routing and a CPE (Configuration in Appendix 3).

The second pilot case was a two CPE backed up internet connection with BGP routing and a CPE (Appendix 4)

The topology with the pilot cases follows TeliaSonera's general customer- and core-network topology shown in figure 22.

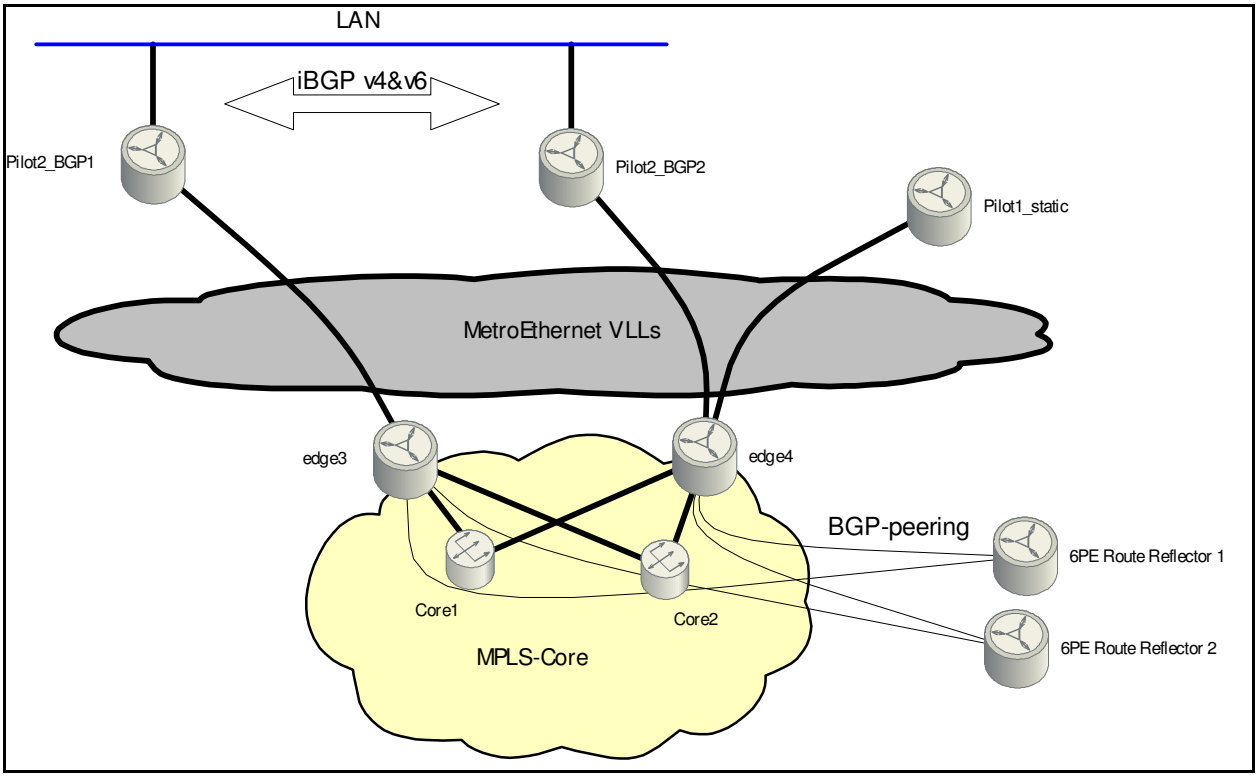


FIGURE 22 customer connection architecture

DNS and other server configurations are out of the scope of this thesis and therefore irrelevant.

7.4.2 Testing

The testing was carried out mostly in the lab network. At first the route reflectors were setup in a private AS (before there was IPv6 connectivity toward peering partners). The first test was to get the BGP-peering up and running toward the lab network. After the BGP was up and running, the IPv6 traffic was routed toward a IPv6-tunnel in order to get initial connectivity to native IPv6 services over the internet.

When the IPv6 connectivity and peering to the peering partners was up, the route reflectors were moved to the production, to AS1759. After that the first production PEs were peered with these route reflectors and the lab network was configured to peer with the PE. When the connectivity was up and running and everything was tested the first pilot customers could be provisioned to the same PE.

8 ROLL-OUT TO PRODUCTION

8.1 Planning and Risk Management

The roll-out project started with planning of the IP-topology. The planning was mostly done using the same planning principles that are used in the rest of the network. New routereflectors for 6PE made it possible to separate the new IPv6 routing from the existing IPv4 network. This way if anything were to go wrong, the problem would not affect the existing production network and the customer connected to it. The edges and route-reflectors for the pilot project were already in the production network. The edges and route-reflectors had a functional software version and therefore did not need upgrading. This way the risk of causing damage to the network in the roll-out phase was minimized. The pilot network topologies were the final topologies.

8.2 Training

Training has two focus points. First, there is the customer provisioning training that focuses mainly on the addressing and the routing in the customer cases, and secondly, there is the PE and Route Reflector configuration.

The customer configuration is based on the same products as in IPv4 and therefore the product specifications do not need to be updated. The training instead will focus on the changes. The training includes a small portion of IPv6 theory; however, it focuses mostly on address reservation and configuration of CPEs.

The training for the PE, Route Reflector and other core-device configuration is mostly hands-on since mostly the IPv6 theoretical knowledge was on a sufficient level. The nature of core device configuration is also quite limited. For example, there are only two route reflectors and new ones might not be needed for years, therefore there is no need for production-processes and –training needed.

8.3 Documentation and Reporting

All configuration and address reservation related material can be found in TeliaSonera's intrawiki. This technical documentation includes most of the details already mentioned in this thesis.

The reporting of address usage does not play a role in IPv6 and therefore it was decided that is not to be addressed, since there are enough addresses.

8.4 Processes and Operational Support Systems

8.4.1 Connection deployment

The connection deployment follows mainly the same routes as with IPv4 except for the address reservation.

The process is as follows:

- A customer orders the connection.
- The physical cable/fiber availability is checked.

- IP planning is done if needed (address reservation etc.)
- Provisioning team configures the edge and access network according to the IP-plan.
- The CPE is installed at customer's site.
- The final configuration is put in to the CPE an PE and the connection is tested.
- The connection is deployed.

8.4.2 Address reservation

Addresses are used for both the customer's LAN and the PE-CPE link. The PE-CPE link addresses are dedicated to each PE and therefore there is no need for separate address reservation. The addresses are formed from the used interfaces and each PE has a unique running number. Each area also has its own link addresses.

The LAN address reservation is done by either the customer or the IP-planning team. The addresses are applied from the Local Internet Registry.

9 RESULTS AND SUMMARY

IPv6 and 6PE were put to production successfully. No major problems were found on the way. The biggest fault in the roll-out was caused by a combination of misconfiguration and a faulty IOS. A great deal has to be done to ensure that the IPv6 connectivity can be offered to other than pure internet customers. For example, the use of 6VPE demands a proper set of features and functionalities from device vendors.

For future interests, the 6PE roll-out is the first step in production networks toward a real IPv6 capable network. With 6PE TeliaSonera is now able to provide real IPv6 connectivity for its internet customers. This connectivity allows several major service providers in Finland to start providing their service for both the IPv4 and IPv6 users natively. Next steps will include the provisioning of IPv6 for consumer customers. The status of this project is not disclosed due to its confidential nature.

The security issues on the operator's point of view can be considered quite straight forward and mostly done with routing. For example TeliaSonera only accepts agreed networks from its customers. TeliaSonera uses blackholing and sinkholing to secure the network and provide customers reliable connectivity. All of this has already been done with IPv4 and was now extended to include IPv6.

The major business implication is that TeliaSonera is now (2010) a major IPv6 network provider in Finland with native IPv6 connectivity to all its peering partners and native IPv6 transit to the rest of the internet through TSIC (TeliaSonera International Carrier) network.

Even though the first phase in the IPv6 provisioning involves only major service provider and large corporation, the rest will follow. If TeliaSonera is able to provide IPv6 connectivity in its network for service providers it is easier to start to offer the same connectivity for the service users. Several matters will have to be thought of. The most significant one will be the real transition from a purely IPv4 environment

via a hybrid environment for IPv4 and IPv6 to a purely IPv6 network. The mechanisms have to be studied and quite possibly several different mechanisms will be used.

People today very dependent on the internet and service and therefore have to start caring about the diminishing amount of IPv4 addresses. Something has to be done. It is not enough that several RFCs are written, they have to be implemented. It is not enough to wait for the market to take care of this. The market is indifferent on the subject of IPv4/IPv6. It will be up to the network engineers and decision makers working for key industries and government agencies to act. The time to act is now, not in the future.

REFERENCES

A Guide to Network Topology. 2008. Accessed on 6.11.2010. http://learn-networking.com/network-design/a-guide-to-network-topology

Bagnulo, M. Matthews, P. & van Beijnum, I. 2010. Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers. Accessed on 6.11.2010. http://tools.ietf.org/html/draft-ietf-behave-v6v4-xlate-stateful-12

Bagnulo, M., Sullivan, A., Matthews, P. & van Beijnum, I. 2010. DNS64: DNS extensions for Network Address Translation from IPv6 Clients. Accessed on 6.11.2010. http://tools.ietf.org/html/draft-ietf-behave-dns64-11

Bates, T., Rekhter, Y. Chandra, R. & Katz, D. 2000. Multiprotocol Extensions for BGP-4. Accessed on 6.11.2010. . http://www.rfc-editor.org/rfc/rfc2462.txt

Bellovin, S. 1996. Defending Against Sequence Number Attacks. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc1948.txt

Coltun, R., Ferguson, D. & Moy, J. 1999. OSPF for IPv6. Accessed on 6.11.2010. . http://www.rfc-editor.org/rfc/rfc2740.txt

Conta, A., Deering, S. & Gupta, M. 2006. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc4443.txt

Convery, S. & Miller, D. IPv6 and IPv4 Threat Comparison and Best-Practice Evaluation (v1.0). Accessed on 6.11.2010. http://seanconvery.com/v6-v4-threats.pdf

De Clercq, J., Ooms, D. Carugi, M. & Le Faucheur, F. 2006. BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc4659.txt

De Clercq, J., Ooms, D., Prevost, S. & Le Faucheur, F. 2007. Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE). Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc4798.txt

De Ghein, L. 2007. MPLS Fundamentals. United States of America: Cisco Press.

Differences Between Multicast and Unicast. Accessed on 6.11.2010. http://support.microsoft.com/kb/291786

FICIX. Accessed on 6.11.2010. http://www.ficix.fi/english/main.php

Finnish Communications Regulatory Authority. Accessed on 6.11.2010. http://www.ficora.fi/en/etusivu.html

Hagen, S. 2006. IPv6 Essentials. Second Edition. United States of America: O'Reilly Media, Inc.

Halabi, S. 2001. Internet Routing Architectures Second Edition. United States of America: Cisco Press.

Hellber, C., Greene, D. & Boyes, T. 2007. Broadband Network Architectures 1st edition. United States of America: Pearson Education, Inc.

Hinden, R. & Deering, S. 1998. IPv6 Multicast Address Assignments. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc2375.txt

Hinden, R. & Deering, S. 1998. Internet Protocol, Version 6 (IPv6) Specification. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc2460.txt

Hinden, R. & Deering, S. 1998. IP Version 6 Addressing Architecture. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc2373.txt

Hinden, R. & Deering, S. 2003. Internet Protocol Version 6 (IPv6) Addressing Architecture. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc3513.txt

Hinden, R. & Deering, S. 2006. IP Version 6 Addressing Architecture. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc4291.txt

Huitema, C. 2006. Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc4380.txt

IANA. IPv4 Multicast Address Space Registry. Accessed on 6.11.2010. http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xml#multicast-addresses-12

Internet Exchanges / Exchange Points / Peering Points (BGP, Border Gateway Protocol / Advanced Internet Routing). Accessed on 6.11.2010. http://www.bgp4.as/internet-exchanges

Internet Protocol – DARPA Internet Program Protocol Specification. 1981. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc791.txt

Johnson, D., Perkins, C. & Arkko, J. 2004. Mobility Support in IPv6. Accessed on 6.11.2010. . http://www.rfc-editor.org/rfc/rfc3775.txt

Kawamura, S. & Kawashima, M. 2010. A recommendation for IPv6 Address Text Representation. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc5952.txt

Kent, S. & Seo, K. 2005. Security Architecture for the Internet Protocol. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc4301.txt

Kim, D., Meyer, D., Kilmer, H. & Farinacci, D. 2003. Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP). Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc3446.txt

Maaniemi, T. 2008. Finnish Network Topology – presentation. TeliaSonera Internal documentation.

McCann, J., Deering, S. & Mogul, J. 1996. Path MTU Discovery for IP version 6. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc1981.txt

Narten, T., Nordmark, E., Simpson, W. & Soliman, H. 1998 Neighbor Discovery for IP version 6 (IPv6). Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc2461.txt

Narten, T., Nordmark, E., Simpson, W. & Soliman, H. 2007 Neighbor Discovery for IP version 6 (IPv6). Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc4861.txt

RIPE Network Coordination Centre. Accessed on 6.11.2010. http://www.ripe.net

Savola, P. & Haberman, B. 2004. Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc3956.txt

Srisuresh, P. & Holdrege, M. 1999. IP Network Address Translator (NAT) Terminology and Considerations. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc2663.txt

Suontausta, P. 2007. TSF IP Core Blackhole Architecture. TeliaSonera Internal documentation.

Thomson, S. &Narten, T. 1998. IPv6 Stateless Address Autoconfiguration. Accessed on 6.11.2010. http://www.rfc-editor.org/rfc/rfc2462.txt

TSF Routing Policy. Accessed on 6.11.2010. https://vilmari.intra.sonera.fi/wiki/index.php/Routing_Policy

APPENDICE

Appendix 1. Route reflector configuration

```
hostname Route-reflector_1_6PE
!
!
!
ip cef
!
!
ip tcp path-mtu-discovery
!
ipv6 unicast-routing
ipv6 cef
no mpls ip
call rsvp-sync
!
key chain ISIS-interface
 key 1
   key-string xxxxxxxxxxxxxx
key chain ISIS-process
 key 1
   key-string xxxxxxxxxxxxxx

!
interface Loopback0
 ip address 10.10.10.1 255.255.255.255
!
interface GigabitEthernet0/1
 mtu 9176
 ip address 10.10.27.1 255.255.255.252
 ip router isis core-igp
 load-interval 30
 carrier-delay msec 50
 media-type gbic
 speed 1000
 duplex full
 no negotiation auto
 no keepalive
 no cdp enable
 clns mtu 1500
 isis circuit-type level-2-only
 isis network point-to-point
 isis metric 160 level-2
 isis authentication mode md5
 isis authentication key-chain ISIS-interface
!
!
router isis core-igp
 net 49.2001.0100.1001.0001.00
 authentication mode md5
 authentication key-chain ISIS-process
 metric-style wide
 set-overload-bit
 spf-interval level-2 10 200 200
 prc-interval 10 200 200
 no hello padding
 log-adjacency-changes
 passive-interface Loopback0
!
router BGP 1759
 BGP router-id 10.10.10.1
 no BGP default ipv4-unicast
 BGP cluster-id 10.10.10.1
 BGP log-neighbor-changes
 timers BGP 30 90 90
 neighbor 6PERR peer-group
 neighbor 6PERR remote-as 1759
 neighbor 6PERR transport path-mtu-discovery
 neighbor 6PERR update-source Loopback0
 neighbor 6PERR version 4
 neighbor 6PERR timers 30 90 90
 neighbor 6PE peer-group
 neighbor 6PE remote-as 1759
```

```
 neighbor 6PE transport path-mtu-discovery
 neighbor 6PE update-source Loopback0
 neighbor 6PE version 4
 neighbor 6PE timers 30 90 90
 neighbor 10.10.12.23 peer-group 6PE
 neighbor 10.10.12.23 description edge1
 neighbor 10.10.12.23 password 7 xxxxxxxxx
 neighbor 10.10.14.49 peer-group 6PE
 neighbor 10.10.14.49 description edge2
 neighbor 10.10.14.49 password 7 xxxxxxxxx
 neighbor 10.10.16.87 peer-group 6PE
 neighbor 10.10.16.87 description edge3
 neighbor 10.10.16.87 password 7 xxxxxxxxx
 neighbor 10.10.16.88 peer-group 6PE
 neighbor 10.10.16.88 description edge3
 neighbor 10.10.16.88 password 7 xxxxxxxxxx
 neighbor 10.10.16.89 peer-group 6PE
 neighbor 10.10.16.89 description edge4
 neighbor 10.10.16.89 password 7 xxxxxxxxxx
 neighbor 10.10.16.20 peer-group 6PE
 neighbor 10.10.16.20 description edge5
 neighbor 10.10.16.20 password 7 xxxxxxxxxx
 neighbor 10.10.25.22 peer-group 6PE
 neighbor 10.10.25.22 description edge6
 neighbor 10.10.25.22 password 7 xxxxxxxxxx
 neighbor 10.10.25.23 peer-group 6PE
 neighbor 10.10.25.23 description edge7
 neighbor 10.10.25.23 password 7 xxxxxxxxxx
 neighbor 10.10.25.24 peer-group 6PE
 neighbor 10.10.25.24 description edge8
 neighbor 10.10.25.24 password 7 xxxxxxxxxx
 neighbor 10.10.25.29 peer-group 6PE
 neighbor 10.10.25.29 description edge9
 neighbor 10.10.25.29 password 7 xxxxxxxxxx
 neighbor 10.10.27.21 peer-group 6PERR
 neighbor 10.10.27.21 description 6PE-RouteReflector
 neighbor 10.10.27.21 password 7 xxxxxxxxxx
 neighbor 10.10.204.37 peer-group 6PE
 neighbor 10.10.204.37 description AS-border1
 neighbor 10.10.204.37 password 7 xxxxxxxxxx
 neighbor 10.10.204.38 peer-group 6PE
 neighbor 10.10.204.38 description AS-border2
 neighbor 10.10.204.38 password 7 xxxxxxxxxx

 !
 address-family ipv6
  neighbor 6PERR send-community
  neighbor 6PERR send-label
  neighbor 6PE send-community
  neighbor 6PE route-reflector-client
  neighbor 6PE send-label
  neighbor 10.10.12.23 activate
  neighbor 10.10.14.49 activate
  neighbor 10.10.16.87 activate
  neighbor 10.10.16.88 activate
  neighbor 10.10.16.89 activate
  neighbor 10.10.16.20 activate
  neighbor 10.10.25.22 activate
  neighbor 10.10.25.23 activate
  neighbor 10.10.25.24 activate
  neighbor 10.10.25.29 activate
  neighbor 10.10.27.21 activate
  neighbor 10.10.204.37 activate
  neighbor 10.10.204.38 activate
  redistribute static route-map v6STATIC
  no synchronization
 exit-address-family
!
ip classless
!
!
ip BGP-community new-format
!
ipv6 route 2001:2000:6000::/40 Null0 tag 333
ipv6 route 2001:2001:6000::/40 Null0 tag 333
ipv6 route 2001:2003::/32 Null0 tag 333
ipv6 route 2001:2060::/32 Null0 tag 333
ipv6 route ::/0 Null0 254
!
!
ipv6 prefix-list v6AGGREGATES seq 10 permit 2001:2060::/32
ipv6 prefix-list v6AGGREGATES seq 20 permit 2001:2000:6000::/40
ipv6 prefix-list v6AGGREGATES seq 30 permit 2001:2001:6000::/40
ipv6 prefix-list v6AGGREGATES seq 40 permit 2001:2003::/32
route-map v6STATIC permit 10
 match tag 333
 match ipv6 address prefix-list v6AGGREGATES
```

```
 set local-preference 160
 set origin igp
 set community 1759:100 1759:103 1759:668
!
route-map v6STATIC deny 20
!
```

Appendix 2: PE-configuration

```
!
hostname edge4
!
!
ipv6 general-prefix p2p-links 2001:2000:600A:xxx::/64
ipv6 general-prefix ma-links 2001:2000:600A:xxx::/56
ipv6 unicast-routing
no ipv6 source-route
!
!

!
key chain ISIS-process
 key 1
   key-string xxxxxxxxxxxxxxx
key chain ISIS-interface
 key 1
   key-string xxxxxxxxxxxxxxx
!
!
ip tcp path-mtu-discovery
!
!
!
!
!
!
interface Null0
 no ip unreachables
 no ipv6 unreachables
!
interface Loopback0
 ip address 10.10.16.89 255.255.255.255
 ipv6 address 2001:2000:6018::0100:1001:6089/128
!
interface TenGigabitEthernet1/1
 description ;physical corelink1
 mtu 9176
 ip address 10.10.16.110 255.255.255.252
 ip router isis core-igp
 mpls ip
 clns mtu 1500
 isis circuit-type level-2-only
 isis network point-to-point
 isis metric 1016 level-2
 isis authentication mode md5
 isis authentication key-chain ISIS-interface
!
interface TenGigabitEthernet2/1
 description physical corelink1
 mtu 9176
 ip address 10.10.16.114 255.255.255.252
 ip router isis core-igp
 mpls ip
 clns mtu 1500
 isis circuit-type level-2-only
 isis network point-to-point
 isis metric 1016 level-2
 isis authentication mode md5
 isis authentication key-chain ISIS-interface
!
interface TenGigabitEthernet3/0/0.104
 description ipv6 labrouter
 encapsulation dot1Q 104
 ip address 10.10.17.197 255.255.255.252
 ipv6 address p2p-links ::3:0:104:1/126
!
interface TenGigabitEthernet3/0/0.2018
 description pilot IPv6
 encapsulation dot1Q 2018
 ipv6 address p2p-links ::3:0:2018:1/126
 ipv6 nd ra suppress
 no ipv6 redirects
!
interface TenGigabitEthernet3/0/0.2019
 description pilot2 BGP
 encapsulation dot1Q 2019
 ipv6 address p2p-links ::3:0:2019:1/126
 ipv6 nd ra suppress
 no ipv6 redirects
end
```

```
!
router isis TSF-core-igp
 net 49.2001.0100.1001.6089.00
 authentication mode md5
 authentication key-chain ISIS-process
 metric-style wide
 spf-interval level-2 10 200 200
 prc-interval 10 200 200
 no hello padding
 log-adjacency-changes
 redistribute maximum-prefix 100
 passive-interface Loopback0
!
router BGP 1759
 no BGP default ipv4-unicast
 BGP log-neighbor-changes
 neighbor RR peer-group
 neighbor RR remote-as 1759
 neighbor RR description AS1759 IPv4 route reflectors
 neighbor RR update-source Loopback0
 neighbor 6PERR peer-group
 neighbor 6PERR remote-as 1759
 neighbor 6PERR transport path-mtu-discovery
 neighbor 6PERR update-source Loopback0
 neighbor 6PERR version 4
 neighbor 6PERR timers 30 90 90
 neighbor v6_TRANSIT_CUSTOMER peer-group
 neighbor v6_TRANSIT_CUSTOMER transport path-mtu-discovery
 neighbor v6_TRANSIT_CUSTOMER version 4
 neighbor v6_TRANSIT_CUSTOMER timers 10 30 30
 neighbor v6_INET_CUSTOMER peer-group
 neighbor v6_INET_CUSTOMER transport path-mtu-discovery
 neighbor v6_INET_CUSTOMER version 4
 neighbor v6_INET_CUSTOMER timers 10 30 30
 neighbor v6_LOCAL_TRANSIT_CUSTOMER peer-group
 neighbor v6_LOCAL_TRANSIT_CUSTOMER transport path-mtu-discovery
 neighbor v6_LOCAL_TRANSIT_CUSTOMER version 4
 neighbor v6_LOCAL_TRANSIT_CUSTOMER timers 10 30 30
 neighbor 2001:2000:600A:xxx:3:0:2019:2 remote-as 65225
 neighbor 2001:2000:600A:xxx:3:0:2019:2 peer-group v6_INET_CUSTOMER
 neighbor 2001:2000:600A:xxx:3:0:2019:2 description pilot2 BGP
 neighbor 10.10.12.13 peer-group RR
 neighbor 10.10.12.13 description core route reflector 1
 neighbor 10.10.12.13 password xxxxxxxxxxxxxxxxxxx
 neighbor 10.10.12.14 peer-group RR
 neighbor 10.10.12.14 description core route reflector 2
 neighbor 10.10.12.14 password xxxxxxxxxxxxxxxxxxx
 neighbor 10.10.10.1 peer-group 6PERR
 neighbor 10.10.10.1 description Route-reflector_1_6PE
 neighbor 10.10.10.1 password xxxxxxxxxxxxxxxxxxx
 neighbor 10.10.10.2 peer-group 6PERR
 neighbor 10.10.10.2 description Route-reflector_2_6PE
 neighbor 10.10.10.2 password xxxxxxxxxxxxxxxxxxx
 !
 address-family ipv4
  no synchronization
  redistribute connected route-map connected_to_BGP
  redistribute static route-map Static-policy-out
  neighbor RR send-community
  neighbor RR next-hop-self
  neighbor RR filter-list 500 in
  neighbor 10.10.12.13 activate
  neighbor 10.10.12.14 activate
  no auto-summary
 exit-address-family
 !
 address-family ipv6
  redistribute connected route-map v6_CONNECTED
  redistribute static route-map v6_STATIC
  no synchronization
  network ::/0
  neighbor 6PERR send-community
  neighbor 6PERR route-map 6PERR-IMPORT in
  neighbor 6PERR route-map 6PERR-EXPORT out
  neighbor 6PERR send-label
  neighbor v6_TRANSIT_CUSTOMER remove-private-as
  neighbor v6_TRANSIT_CUSTOMER route-map v6_Global-Transit-policy-in in
  neighbor v6_TRANSIT_CUSTOMER route-map v6_Global-Transit-policy-out out
  neighbor v6_TRANSIT_CUSTOMER maximum-prefix 10 restart 10
  neighbor v6_LOCAL_TRANSIT_CUSTOMER remove-private-as
  neighbor v6_LOCAL_TRANSIT_CUSTOMER route-map v6_Local-Transit-policy-in in
  neighbor v6_LOCAL_TRANSIT_CUSTOMER route-map v6_Local-Transit-policy-out out
  neighbor 2001:2000:600A:xxx:3:0:2019:2 activate
  neighbor 10.10.10.1 activate
  neighbor 10.10.10.2 activate
 exit-address-family
!
```

```
ip BGP-community new-format
ip community-list standard Blackhole permit 1759:666
ip community-list standard Set-LocalPref-50 permit 1759:1050
ip community-list standard Set-LocalPref-110 permit 1759:1110
ip community-list standard Set-LocalPref-150 permit 1759:1150
ip community-list standard Suppress-any-peer permit 1759:1000
ip community-list standard Customer-peer permit 1759:100
ip community-list standard CIX-peer permit 1759:110
ip community-list standard Upstream-peer permit 1759:130
ip community-list standard Private-peer permit 1759:120
ip community-list expanded Customer-RP-communities permit (^|_)1759:1[0-3][0-9](_|$)
ip community-list expanded Global-Transit-policy-x1 permit
(^|_)1759:100(_|_.*_)1759:1001(_|$)
ip community-list expanded Global-Transit-policy-x1 permit
(^|_)1759:110(_|_.*_)1759:1001(_|$)
ip community-list expanded Global-Transit-policy-x1 permit
(^|_)1759:120(_|_.*_)1759:1001(_|$)
ip community-list expanded Global-Transit-policy-x1 permit
(^|_)1759:130(_|_.*_)1759:1001(_|$)
ip community-list expanded Global-Transit-policy-x2 permit
(^|_)1759:100(_|_.*_)1759:1002(_|$)
ip community-list expanded Global-Transit-policy-x2 permit
(^|_)1759:110(_|_.*_)1759:1002(_|$)
ip community-list expanded Global-Transit-policy-x2 permit
(^|_)1759:120(_|_.*_)1759:1002(_|$)
ip community-list expanded Global-Transit-policy-x2 permit
(^|_)1759:130(_|_.*_)1759:1002(_|$)
ip community-list expanded Global-Transit-policy-x3 permit
(^|_)1759:100(_|_.*_)1759:1003(_|$)
ip community-list expanded Global-Transit-policy-x3 permit
(^|_)1759:110(_|_.*_)1759:1003(_|$)
ip community-list expanded Global-Transit-policy-x3 permit
(^|_)1759:120(_|_.*_)1759:1003(_|$)
ip community-list expanded Global-Transit-policy-x3 permit
(^|_)1759:130(_|_.*_)1759:1003(_|$)
ip community-list expanded Local-Transit-policy-x1 permit
(^|_)1759:100(_|_.*_)1759:1001(_|$)
ip community-list expanded Local-Transit-policy-x2 permit
(^|_)1759:100(_|_.*_)1759:1002(_|$)
ip community-list expanded Local-Transit-policy-x3 permit
(^|_)1759:100(_|_.*_)1759:1003(_|$)
ip community-list expanded Blackholes permit (^|_)1759:66[6-8](_|$)
ip community-list expanded Sinkhole permit (^|_)1759:669(_|$)
ip as-path access-list 100 permit (^|_)((645[1-9][2-9])|(64[6-9][0-9][0-9])|(65[0-
9][0-9][0-9]))(_|$)
ip as-path access-list 500 deny 64619
ip as-path access-list 500 permit .*
no ip http server
no ip http secure-server
!
ipv6 route 2001:2000:600A:xxx::/56 Null0
ipv6 route 2001:2000:6018:FFFF:FFFF:FFFF:FFFF:FFFF/128 Null0
ipv6 route 2001:2001:6000::/52 2001:2000:600A:xxx:3:0:104:2 tag 333
ipv6 route 2001:2001:6000:8000::/52 2001:2000:600A:xxx:3:0:104:2 tag 333
ipv6 route 2001:2060:40::/48 2001:2000:600A:xxx:3:0:104:2 tag 333
ipv6 route 2001:2060:45::/48 2001:2000:600A:xxx:3:0:2018:2 tag 333
!
!
ipv6 prefix-list v6_CONNECTED seq 5 permit 2001:2000:6018::0100:1001:6089/128
ipv6 prefix-list v6_CONNECTED seq 10 deny ::/0 le 128
!
ipv6 prefix-list v6_CUSTOMER_NETS seq 5 permit 2001:2060::/32 ge 33
!
ipv6 prefix-list v6_CUSTOMER_PI_TO_BGP seq 10 permit ::/0
ipv6 prefix-list v6_CUSTOMER_PI_TO_BGP seq 1000 deny ::/0 le 128
!
ipv6 prefix-list v6_DEFAULT seq 5 permit ::/0
!
ipv6 prefix-list v6_DEFAULT_Pilot_BGP seq 5 permit ::/0
!
ipv6 prefix-list v6_STATIC seq 5 permit 2001:2000:600A:xxx::/56
ipv6 prefix-list v6_STATIC seq 10 deny 2001:2000:6000::/39 ge 40
!
ipv6 prefix-list v6_black_sinkhole seq 5 permit
2001:2000:6018:FFFF:FFFF:FFFF:FFFF:FFFF/128
ipv6 prefix-list v6_black_sinkhole seq 10 permit
2001:2000:6018:FFFF:FFFF:FFFF:FFFF:FFFE/128
!
ipv6 prefix-list v6_deny_short_and_long_prefixes seq 10 permit ::/0 le 17
ipv6 prefix-list v6_deny_short_and_long_prefixes seq 20 permit ::/0 ge 49
!
ipv6 prefix-list v6connected seq 5 permit 2001:2000:6018::0100:1001:6089/128
!
ipv6 prefix-list v6static seq 5 permit 2001:2000:600A:xxx::/56
ipv6 prefix-list v6static seq 10 permit ::/0
route-map v6_STATIC permit 10
 match source-protocol static
```

```
 match ipv6 address prefix-list v6_STATIC
 set community 1759:1000
!
route-map v6_STATIC permit 20
 match tag 333
 match source-protocol static
 match ipv6 address prefix-list v6_CUSTOMER_NETS
 set community 1759:1000
!
route-map v6_STATIC permit 30
 match tag 333
 match source-protocol static
 match ipv6 address prefix-list v6_CUSTOMER_PI_TO_BGP
 set community 1759:100 1759:103
!
route-map 6PERR-EXPORT permit 10
 set community 1759:30010 additive
!
route-map connected_to_BGP permit 10
 description allow connected to BGP
 match ip address connected_to_BGP
!
route-map connected_to_BGP permit 17
 match ip address do_not_advertise_any_peer
 set community 1759:1000
!
route-map connected_to_BGP deny 20
 description deny rest
!
route-map 6PERR-IMPORT permit 10
 match community Blackholes
 set ipv6 next-hop 2001:2000:6018:FFFF:FFFF:FFFF:FFFF:FFFF
!
route-map 6PERR-IMPORT permit 20
 match community Sinkhole
 set ipv6 next-hop 2001:2000:6018:FFFF:FFFF:FFFF:FFFF:FFFE
!
route-map 6PERR-IMPORT permit 30
!
route-map Global-redundant-primary-in permit 10
 description Global-redundant-primary
 set local-preference 160
 set community 1759:100 1759:103 additive
!
route-map Default_to_Pilot_BGP permit 10
 match ipv6 address prefix-list v6_DEFAULT_Pilot_BGP
!
route-map Local-Transit-policy-in permit 10
 description Set-Default-LocalPref
 continue 20
 set local-preference 160
!
route-map Local-Transit-policy-in deny 20
 description Remove-private-AS
 match as-path 100
!
route-map Local-Transit-policy-in permit 30
 description Blackhole
 match community Blackhole
 set ip next-hop x.x.x.x
!
route-map Local-Transit-policy-in permit 40
 description Set-LocalPref-50
 match community Set-LocalPref-50
 continue 70
 set local-preference 50
!
route-map Local-Transit-policy-in permit 50
 description Set-LocalPref-110
 match community Set-LocalPref-110
 continue 70
 set local-preference 110
!
route-map Local-Transit-policy-in permit 60
 description Set-LocalPref-150
 match community Set-LocalPref-150
 continue 70
 set local-preference 150
!
route-map Local-Transit-policy-in permit 70
 description Default-action
 set comm-list Customer-RP-communities delete
 set community 1759:100 1759:101 additive
!
route-map Only-default-route permit 10
 description Announce only default-route
 match ip address Default-route
```

```
!
route-map Only-default-route deny 20
!
route-map Global-Transit-policy-out deny 10
 description Remove-long-prefixes
 match ip address Longer-prefixes
!
route-map Global-Transit-policy-out deny 20
 description Remove-customer-suppressed
 match community Suppress-any-peer
!
route-map Global-Transit-policy-out permit 30
 description Prepend-x1
 match community Global-Transit-policy-x1
 continue 60
 set as-path prepend 1759
!
route-map Global-Transit-policy-out permit 40
 description Prepend-x2
 match community Global-Transit-policy-x2
 continue 60
 set as-path prepend 1759 1759
!
route-map Global-Transit-policy-out permit 50
 description Prepend-x3
 match community Global-Transit-policy-x3
 continue 60
 set as-path prepend 1759 1759 1759
!
route-map Global-Transit-policy-out permit 60
 description Accept-valid-RP-community
 match community Customer-peer CIX-peer Private-peer Upstream-peer
!
route-map Global-Transit-policy-out deny 70
 description Default-action
!
route-map v6_Global-Transit-policy-out deny 10
 description v6-remove-short-and-long-prefixes
 match ipv6 address prefix-list v6_deny_short_and_long_prefixes
!
route-map v6_Global-Transit-policy-out deny 20
 description remove-customer-suppressed
 match community Suppress-any-peer
!
route-map v6_Global-Transit-policy-out permit 30
 description Prepend-x1
 match community Global-Transit-policy-x1
 continue 60
 set as-path prepend 1759
!
route-map v6_Global-Transit-policy-out permit 40
 description Prepend-x2
 match community Global-Transit-policy-x2
 continue 60
 set as-path prepend 1759 1759
!
route-map v6_Global-Transit-policy-out permit 50
 description Prepend-x3
 match community Global-Transit-policy-x3
 continue 60
 set as-path prepend 1759 1759 1759
!
route-map v6_Global-Transit-policy-out permit 60
 description Accept-valid-RP-community
 match community Customer-peer CIX-peer Private-peer Upstream-peer
!
route-map v6_Global-Transit-policy-out deny 70
 description Default-action
!
route-map loopback-remove deny 10
 match ip address prefix-list loopback-filter
!
route-map loopback-remove permit 20
!
route-map v6connected permit 10
 match ipv6 address prefix-list v6connected
!
route-map v6_Local-Transit-policy-in permit 10
 description Set-Default-LocalPref
 continue 20
 set local-preference 160
!
route-map v6_Local-Transit-policy-in deny 20
 description Remove-private-AS
 match as-path 100
!
route-map v6_Local-Transit-policy-in permit 30
```

```
 description v6_Blackhole
 match community Blackhole
 set ipv6 next-hop 2001:2000:6018:FFFF:FFFF:FFFF:FFFF:FFFF
!
route-map v6_Local-Transit-policy-in permit 40
 description Set-LocalPref-50
 match community Set-LocalPref-50
 continue 70
 set local-preference 50
!
route-map v6_Local-Transit-policy-in permit 50
 description Set-LocalPref-110
 match community Set-LocalPref-110
 continue 70
 set local-preference 110
!
route-map v6_Local-Transit-policy-in permit 60
 description Set-LocalPref-150
 match community Set-LocalPref-150
 continue 70
 set local-preference 150
!
route-map v6_Local-Transit-policy-in permit 70
 description Default-action
 set comm-list Customer-RP-communities delete
 set community 1759:100 1759:101 additive
!
route-map v6_CONNECTED permit 10
 match ipv6 address prefix-list v6_CONNECTED
 set community 1759:1000
!
route-map Global-Transit-policy-in permit 10
 description Set-Default-LocalPref
 continue 20
 set local-preference 160
!
route-map Global-Transit-policy-in deny 20
 description Remove-private-AS
 match as-path 100
!
route-map Global-Transit-policy-in permit 30
 description Blackhole
 match community Blackhole
 set ip next-hop 192.168.0.1
!
route-map Global-Transit-policy-in permit 40
 description Set-LocalPref-50
 match community Set-LocalPref-50
 continue 70
 set local-preference 50
!
route-map Global-Transit-policy-in permit 50
 description Set-LocalPref-110
 match community Set-LocalPref-110
 continue 70
 set local-preference 110
!
route-map Global-Transit-policy-in permit 60
 description Set-LocalPref-150
 match community Set-LocalPref-150
 continue 70
 set local-preference 150
!
route-map Global-Transit-policy-in permit 70
 description Default-action
 set comm-list Customer-RP-communities delete
 set community 1759:100 1759:103 additive
!
route-map Local-Transit-policy-out deny 10
 description Remove-long-prefixes
 match ip address Longer-prefixes
!
route-map Local-Transit-policy-out deny 20
 description Remove-customer-suppressed
 match community Suppress-any-peer
!
route-map Local-Transit-policy-out permit 30
 description Prepend-x1
 match community Local-Transit-policy-x1
 continue 60
 set as-path prepend 1759
!
route-map Local-Transit-policy-out permit 40
 description Prepend-x2
 match community Local-Transit-policy-x2
 continue 60
 set as-path prepend 1759 1759
```

```
!
route-map Local-Transit-policy-out permit 50
 description Prepend-x3
 match community Local-Transit-policy-x3
 continue 60
 set as-path prepend 1759 1759 1759
!
route-map Local-Transit-policy-out permit 60
 description Accept-valid-RP-community
 match community Customer-peer
!
route-map Local-Transit-policy-out deny 70
 description Default-action
!
route-map No-default-redistribute permit 10
 description No default-route redistribution outside of own AS
 set community no-export
!
route-map ISP-default-out permit 10
 match ip address prefix-list DEFAULT
!
route-map Static-policy-out permit 10
 description Redistribute with Global-Transit-policy community, LocalPref 160
 match tag 103
 set local-preference 160
 set community 1759:100 1759:103
!
route-map Static-policy-out permit 20
 description Redistribute with National-Transit-policy community, LocalPref 160
 match tag 102
 set local-preference 160
 set community 1759:100 1759:102
!
route-map Static-policy-out permit 30
 description Redistribute with Local-Transit-policy community, LocalPref 160
 match tag 101
 set local-preference 160
 set community 1759:100 1759:101
!
route-map Static-policy-out permit 40
 description Redistribute with Global-Transit-policy community, LocalPref 150
 match tag 103150
 set local-preference 150
 set community 1759:100 1759:103
!
route-map Static-policy-out permit 50
 description Redistribute with Local-Transit-policy community, LocalPref 150
 match tag 102150
 set local-preference 150
 set community 1759:100 1759:102
!
route-map Static-policy-out permit 60
 description Redistribute with Local-Transit-policy community, LocalPref 150
 match tag 101150
 set local-preference 150
 set community 1759:100 1759:101
!
route-map Static-policy-out permit 70
 description Redistribute with Suppress community, LocalPref 150
 match tag 150
 set local-preference 150
 set community 1759:100 1759:1000
!
route-map Static-policy-out permit 80
 description Default redistibution, Suppress community and LocalPref 160
 set local-preference 160
 set community 1759:100 1759:1000
!
route-map v6_Local-Transit-policy-out deny 10
 description v6-remove-short-and-long-prefixes
 match ipv6 address prefix-list v6_deny_short_and_long_prefixes
!
route-map v6_Local-Transit-policy-out deny 20
 description remove-customer-suppressed
 match community Suppress-any-peer
!
route-map v6_Local-Transit-policy-out permit 30
 description Prepend-x1
 match community Local-Transit-policy-x1
 continue 60
 set as-path prepend 1759
!
route-map v6_Local-Transit-policy-out permit 40
 description Prepend-x2
 match community Local-Transit-policy-x2
 continue 60
 set as-path prepend 1759 1759
```

```
!
route-map v6_Local-Transit-policy-out permit 50
 description Prepend-x3
 match community Local-Transit-policy-x3
 continue 60
 set as-path prepend 1759 1759 1759
!
route-map v6_Local-Transit-policy-out permit 60
 description Accept-valid-RP-community
 match community Customer-peer
!
route-map v6_Local-Transit-policy-out deny 70
 description Default-action
!
route-map v6static permit 10
 match ipv6 address prefix-list v6static
!
route-map v6_Global-Transit-policy-in permit 10
 description Set-Default-LocalPref
 continue 20
 set local-preference 160
!
route-map v6_Global-Transit-policy-in deny 20
 description Remove-private-AS
 match as-path 100
!
route-map v6_Global-Transit-policy-in permit 30
 description v6_Blackhole
 match community Blackhole
 set ipv6 next-hop 2001:2000:6018:FFFF:FFFF:FFFF:FFFF:FFFF
!
route-map v6_Global-Transit-policy-in permit 40
 description Set-LocalPref-50
 match community Set-LocalPref-50
 continue 70
 set local-preference 50
!
route-map v6_Global-Transit-policy-in permit 50
 description Set-LocalPref-110
 match community Set-LocalPref-110
 continue 70
 set local-preference 110
!
route-map v6_Global-Transit-policy-in permit 60
 description Set-LocalPref-150
 match community Set-LocalPref-150
 continue 70
 set local-preference 150
!
route-map v6_Global-Transit-policy-in permit 70
 description Default-action
 set comm-list Customer-RP-communities delete
 set community 1759:100 1759:103 additive
!
mpls ldp router-id Loopback0
!
ipv6 access-list v6_NODE_ACCESS
 deny ipv6 any any
!

line vty 0 4
 ipv6 access-class v6_NODE_ACCESS in
 transport input lat pad udptn telnet rlogin ssh acercon
```

Appendix 3: CPE-configuration (non-BGP Cisco)

```
hostname pilot1 IPv6
!
ip cef
!
!
ipv6 unicast-routing
ipv6 cef
multilink bundle-name authenticated
!
!
!
!
interface FastEthernet0
 bandwidth 10240
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0.100
 encapsulation dot1Q 100
 ip address 10.14.127.181 255.255.255.252
!
interface FastEthernet0.101
 encapsulation dot1Q 101
 ipv6 address 2001:2000:600A:xxx:3:0:2018:2/126
!

interface Vlan1
 ip address 192.168.1.1 255.255.255.224
 ipv6 address xxxx::C0A8:DE01 link-local
!
ip route 0.0.0.0 0.0.0.0 10.14.127.182
!
!
no ip http server
no ip http secure-server
!
!
ipv6 route 2001:2060:45::/48 Vlan1 xxxx::C0A8:DE02
ipv6 route ::/0 2001:2000:600A:xxx:3:0:2018:1
!
!
!
!
ipv6 access-list IPv6ANTISPOOF
 permit ipv6 2001:2060:45::/48 any
 permit ipv6 xxxx::C0A8:DE00/126 any
 deny ipv6 any any
!
control-plane


!

end
```

Appendix 4: CPE configuration BGP (Juniper)

```
system {
    host-name pilot2 IPv6 BGP;

interfaces {
    ge-0/0/0 {
        description TO_LAN;
        unit 0 {
            family inet {
                filter {
                    input IS00083672-CLASSIFYING;
                }
                address 192.168.71.130/25 {
                    vrrp-group 1 {
                        virtual-address 192.16871.129;
                        priority 192;
                        preempt;
                        accept-data;
                        track {
                            interface ge-3/0/0 {
                                priority-cost 96;
                            }
                        }
                    }
                }
            }
            family inet6 {
                address 2001:2060:0042:xxxx::3/124 {
                    vrrp-inet6-group 6 {
                        virtual-inet6-address 2001:2060:0042:xxxx::1;
                        virtual-link-local-address xxxx::200:5e00:3e00:0200;
                        priority 192;
                        preempt;
                        accept-data;
                        track {
                            interface ge-3/0/0 {
                                priority-cost 96;
                            }
                        }
                    }
                }
                address xxxx::221:0024:dc04:eb01/64;
            }
        }
    }
    ge-3/0/0 {
        description management;
        per-unit-scheduler;
        vlan-tagging;
        unit 0 {
            vlan-id 100;
            family inet {
                address 10.41.120.197/30;
            }
        }
        unit 110 {
            vlan-id 110;
            family inet6 {
                address 2001:2000:600A:xxx:3:0:2019:2/126;
            }
        }
    }
}
routing-options {
    rib inet6.0 {
        static {
            route 2001:2060:0042:8000::/49 next-hop 2001:2060:0042:xxxx::6;
        }
    }
    }
}
protocols {
    router-advertisement {
        interface ge-0/0/0.0 {
            virtual-router-only;
        }
    }
    BGP {
        local-as 65225;
        group SONERA {
            type external;
            neighbor 10.41.120.198 {
                local-address 10.41.120.197;
                peer-as 12582;
```

```
            }
        }
        group iBGP_to_backup_IPv4 {
            type internal;
            local-as 65225;
            neighbor 192.168.71.131 {
                local-address 192.168.71.130;
                hold-time 30;
            }
        }
        group IPv6_To_edge4 {
            type external;
            export IPV6_ROUTES;
            peer-as 1759;
            neighbor 2001:2000:600A:xxx:3:0:2019:1;
        }
        group ipv6_IBGP_to_backup {
            type internal;
            neighbor 2001:2060:0042:xxxx::2;
        }
    }
}
policy-options {
    policy-statement DEFAULT_only {
        term Default {
            from {
                protocol BGP;
                route-filter ::/0 exact;
            }
            then {
                local-preference 200;
                accept;
            }
        }
        term reject {
            then reject;
        }
    }
    policy-statement IPV6_ROUTES {
        term static {
            from {
                protocol static;
                route-filter 2001:2060:0042:8000::/49 exact;
            }
            then accept;
        }
        term default_action {
            then reject;
        }
    }
    community 12582:200 members 12582:200;
    community 65225:154 members 65225:154;
    as-path no_transit "()";
}
```

Appendix 5: Finnish Network